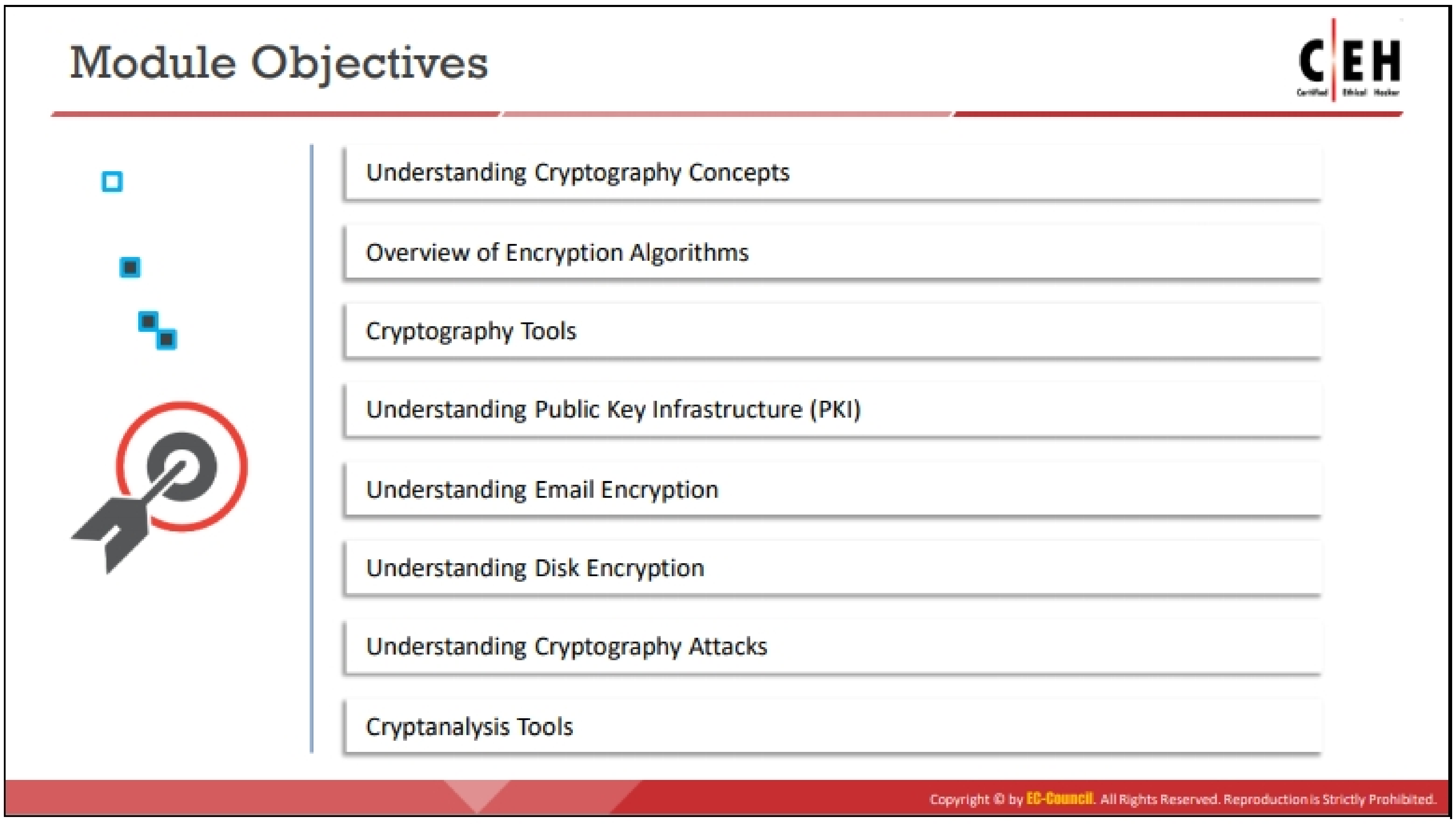
C|EH™

Certified Ethical Hacker

Module 20:

Cryptography



Module Objectives

- Understanding Cryptography Concepts
- Overview of Encryption Algorithms
- Cryptography Tools
- Understanding Public Key Infrastructure (PKI)
- Understanding Email Encryption
- Understanding Disk Encryption
- Understanding Cryptography Attacks
- Cryptanalysis Tools

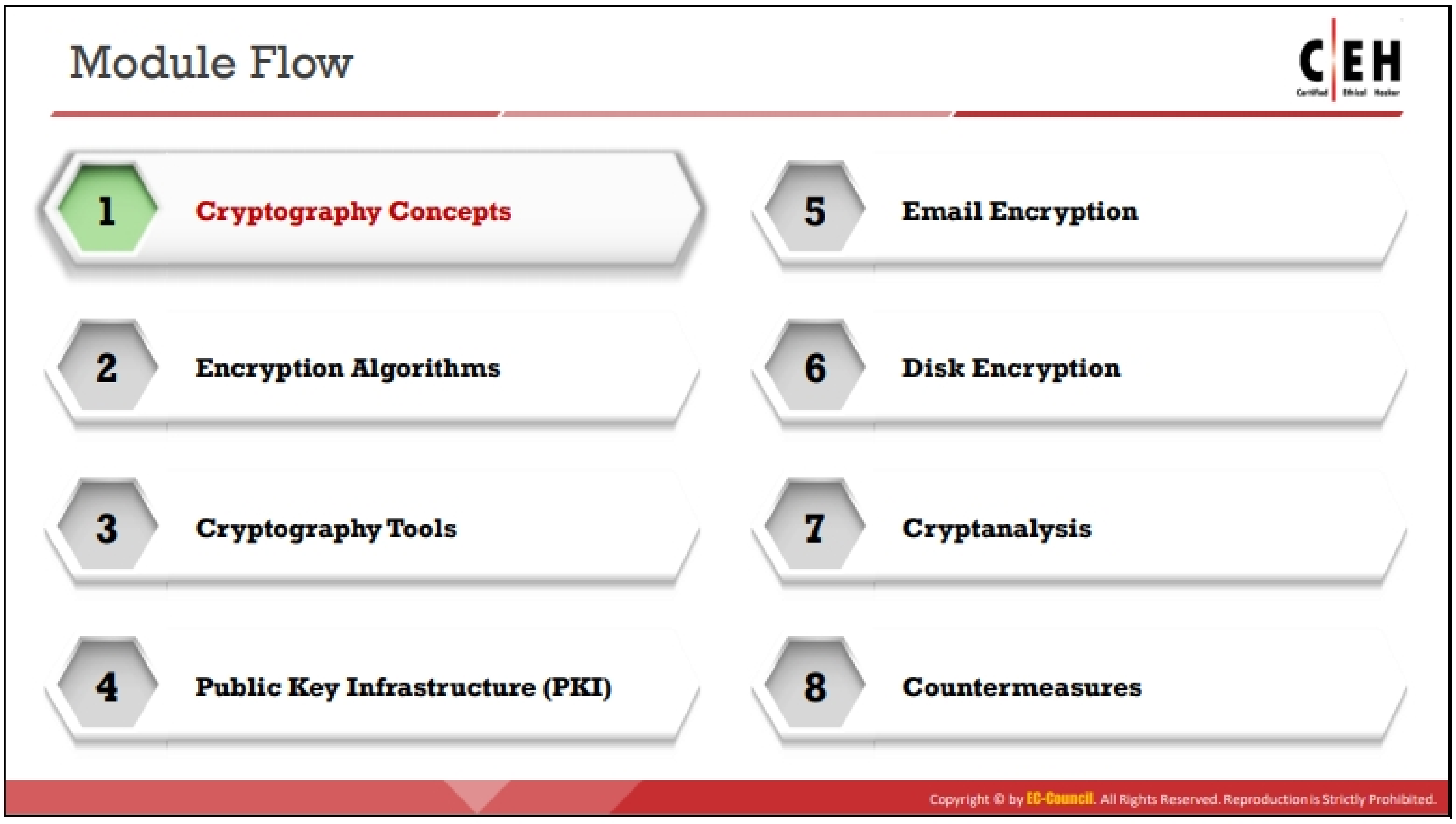
Module Objectives

With the increasing adoption of the Internet (World Wide Web) for business and personal communication, securing sensitive information such as credit card details, PINs, bank account numbers, and private messages is becoming increasingly important, albeit more difficult to achieve. Today's information-based organizations extensively use the Internet for e-commerce, market research, customer support, and a variety of other activities. Data security is critical to online business and communication privacy.

Cryptography and cryptographic ("crypto") systems help in securing data against interception and compromise during online transmissions. This module provides a comprehensive understanding of different cryptosystems and algorithms, one-way hash functions, public-key infrastructures (PKIs), and the different ways in which cryptography can ensure the privacy and security of online communication. It also covers various tools used to encrypt sensitive data.

At the end of this module, you will be able to

- Describe cryptography concepts
- Understand different encryption algorithms
- Use different cryptography tools
- Describe public key infrastructure (PKI)
- Apply email encryption
- Apply disk encryption
- Describe various cryptography attacks
- Use different cryptanalysis tools

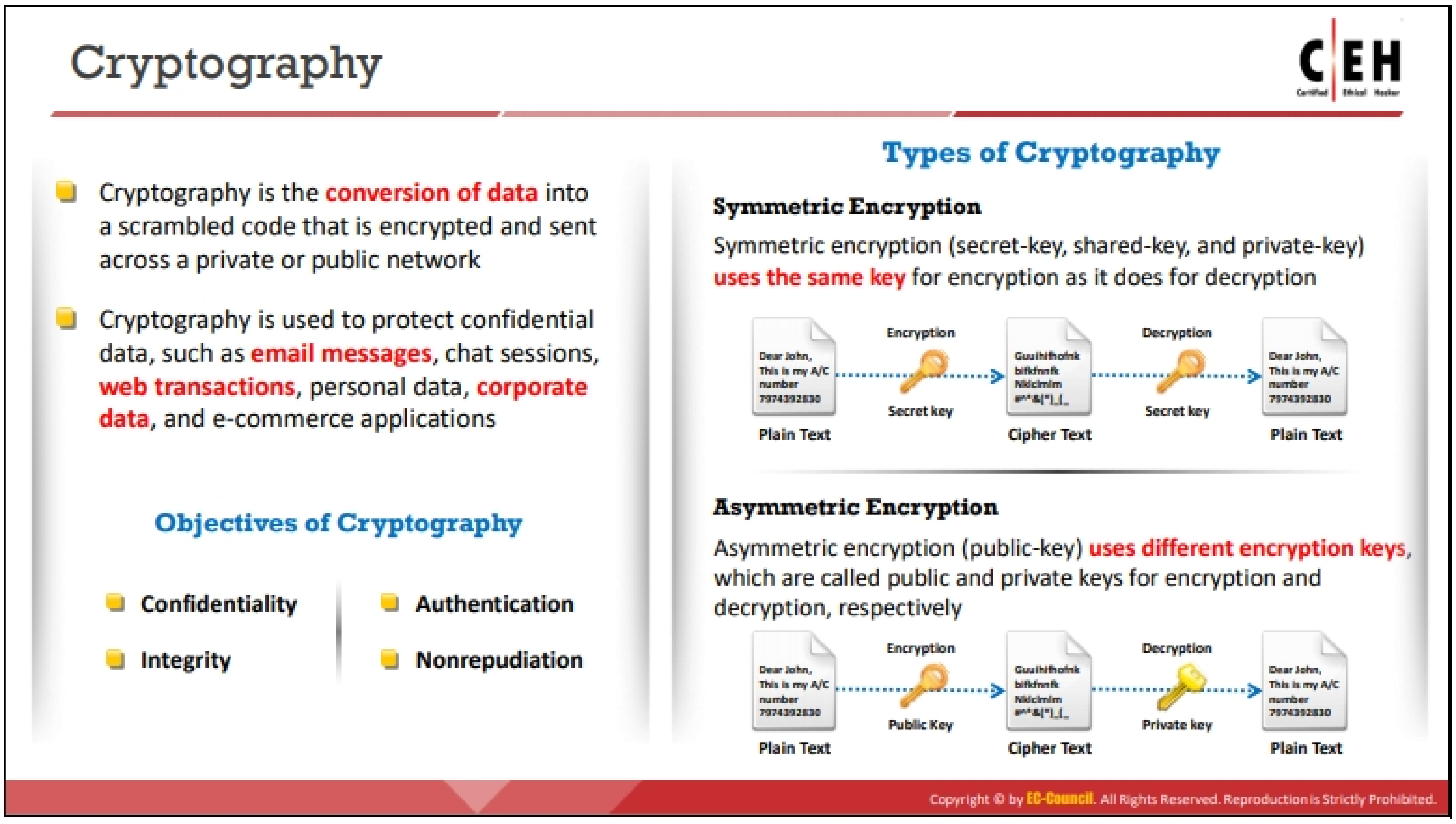


Module Flow

1. **Cryptography Concepts**
2. **Encryption Algorithms**
3. **Cryptography Tools**
4. **Public Key Infrastructure (PKI)**
5. **Email Encryption**
6. **Disk Encryption**
7. **Cryptanalysis**
8. **Countermeasures**

Cryptography Concepts

Cryptography enables one to secure transactions, communications, and other processes performed in the electronic world. This section deals with cryptography and its associated concepts, which will enable you to understand the advanced topics covered later in this module.



Cryptography

- Cryptography is the **conversion of data** into a scrambled code that is encrypted and sent across a private or public network
- Cryptography is used to protect confidential data, such as **email messages**, chat sessions, **web transactions**, personal data, **corporate data**, and e-commerce applications

Objectives of Cryptography

- Confidentiality
- Integrity
- Authentication
- Nonrepudiation

Types of Cryptography

Symmetric Encryption

Symmetric encryption (secret-key, shared-key, and private-key) **uses the same key** for encryption as it does for decryption

Asymmetric Encryption

Asymmetric encryption (public-key) **uses different encryption keys**, which are called public and private keys for encryption and decryption, respectively

Cryptography

"Cryptography" comes from the Greek words ***kryptos***, meaning "concealed, hidden, veiled, secret, or mysterious," and ***graphia***, meaning "writing"; thus, cryptography is "the art of secret writing."

Cryptography is the practice of concealing information by converting plaintext (readable format) into ciphertext (unreadable format) using a key or encryption scheme. It is the process of converting data into a scrambled code that is encrypted and sent across a private or public network. Cryptography protects confidential data such as email messages, chat sessions, web transactions, personal data, corporate data, e-commerce applications, and many other types of communication. Encrypted messages can, at times, be decrypted by cryptanalysis (code breaking), even though modern encryption techniques are virtually unbreakable.

Objectives of Cryptography

- **Confidentiality**: Assurance that the information is accessible only to those authorized to access it.
- **Integrity**: Trustworthiness of data or resources in terms of preventing improper and unauthorized changes.
- **Authentication**: Assurance that the communication, document, or data is genuine.
- **Nonrepudiation**: Guarantee that the sender of a message cannot later deny having sent the message and that the recipient cannot deny having received the message.

Cryptography Process

Plaintext (readable format) is encrypted by means of encryption algorithms such as RSA, DES, and AES, resulting in a ciphertext (unreadable format) that, on reaching the destination, is decrypted into readable plaintext.

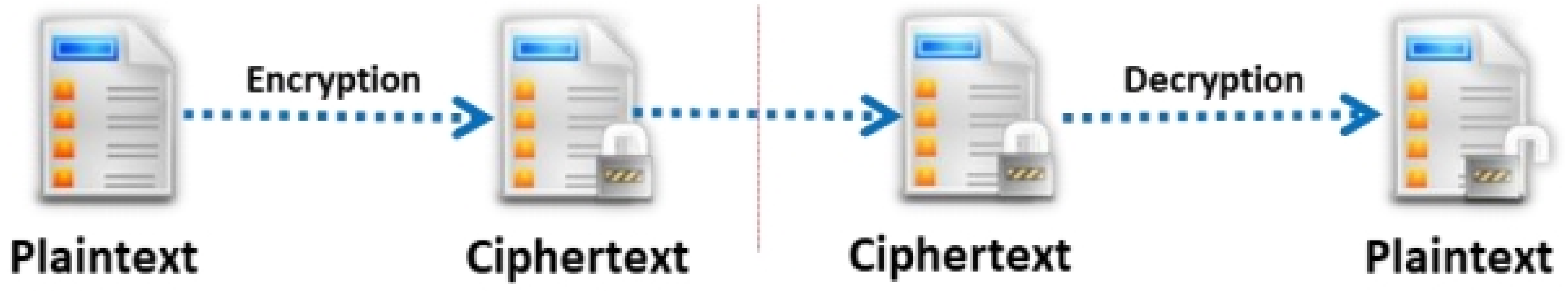

Encryption
Decryption
Plaintext
Ciphertext
Ciphertext
Plaintext

Figure 20.1: Example of Cryptography

Types of Cryptography

Cryptography is categorized into two types according to the number of keys employed for encryption and decryption:

- **Symmetric Encryption**

  Symmetric encryption requires that both the sender and the receiver of the message possess the same encryption key. The sender uses a key to encrypt the plaintext and sends the resultant ciphertext to the recipient, who uses the same key (used for encryption) to decrypt the ciphertext into plaintext. Symmetric encryption is also known as secret-key cryptography, as it uses only one secret key to encrypt and decrypt the data. This type of cryptography works well when you are communicating with only a few people.

  Because the sender and receiver must share the key before sending any messages, this technique is of limited use for the Internet, where individuals who have not had prior contact frequently require a secure means of communication. The solution to this problem is asymmetric encryption (public-key cryptography).

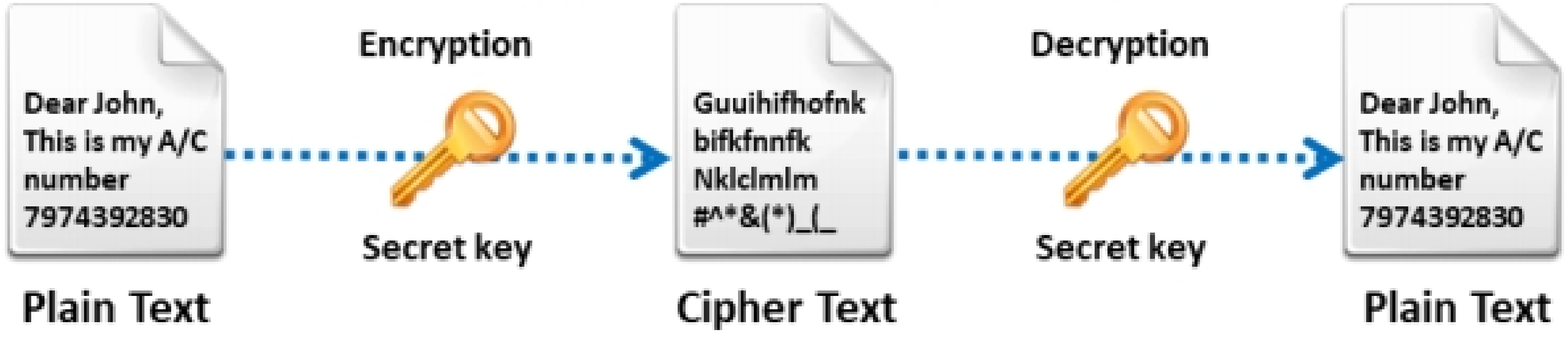

Encryption
Decryption
Dear John,
This is my A/C
number
7974392830
Secret key
Guuihifhofnk
bifkfnnfk
Nklclmlm
#^*&(*)_(_
Secret key
Dear John,
This is my A/C
number
7974392830
Plain Text
Cipher Text
Plain Text

Figure 20.2: Symmetric Encryption

- **Asymmetric Encryption**

  The concept of asymmetric encryption (also known as public-key cryptography) was introduced to solve key-management problems. Asymmetric encryption involves both a public key and a private key. The public key is publicly available, whereas the sender keeps the private key secret.

  An asymmetric-key system is an encryption method that uses a key pair comprising a public key available to anyone and a private key held only by the key owner, which helps to provide confidentiality, integrity, authentication, and nonrepudiation in data management.

Asymmetric encryption uses the following sequence to send a message:

1. An individual finds the public key of the person he or she wants to contact in a directory.
2. This public key is used to encrypt a message that is then sent to the intended recipient.
3. The receiver uses the private key to decrypt the message and reads it.

No one but the holder of the private key can decrypt a message encrypted with the corresponding public key. This increases the security of the information because all communications involve only public keys; the message sender never transmits or shares the private keys. The sender must link public keys with usernames in a secure manner to ensure that individuals claiming to be the intended recipient do not intercept the information. To meet the need for authentication, one can use digital signatures.

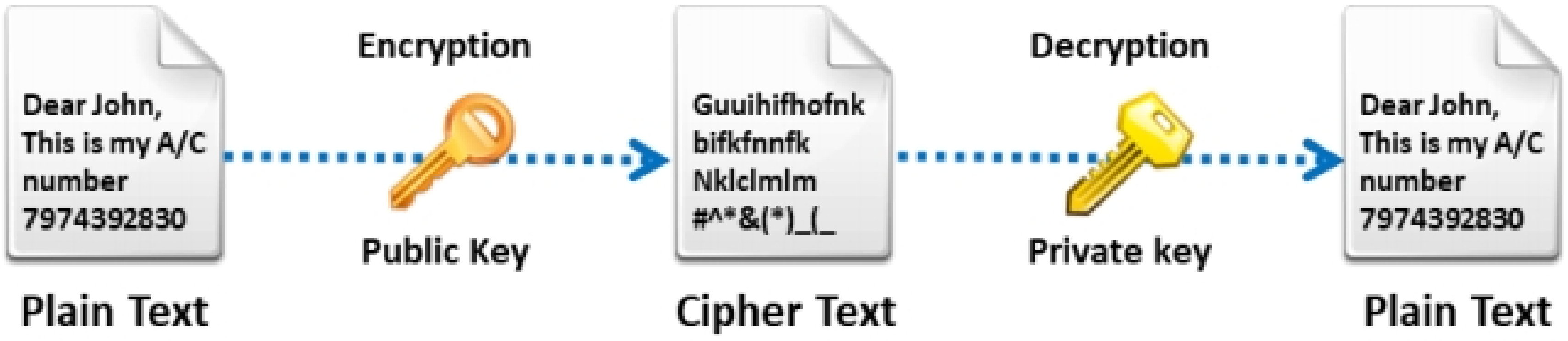


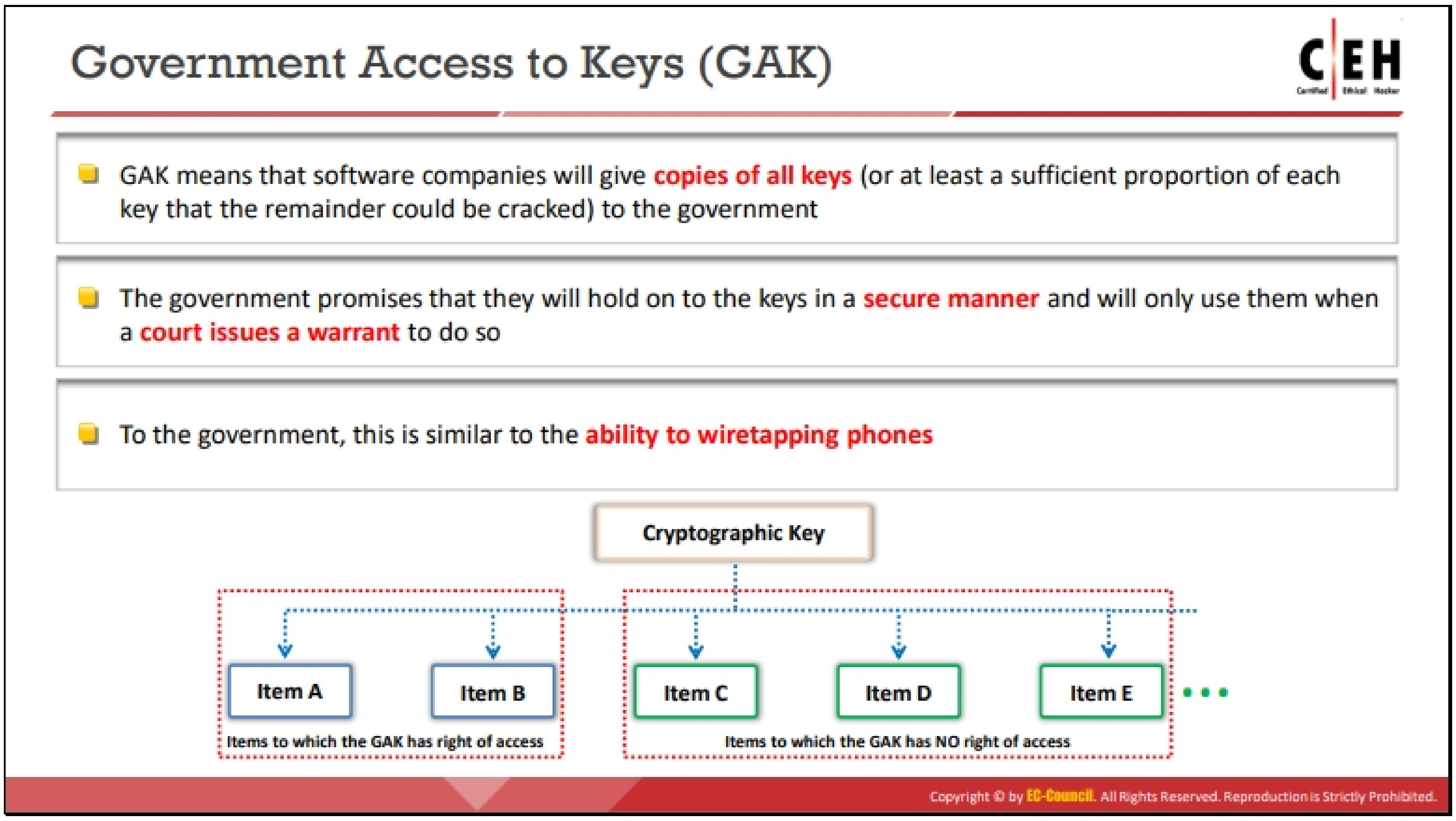
Figure 20.3: Asymmetric Encryption

Strengths and Weaknesses of Crypto Methods

	Symmetric Encryption	Asymmetric Encryption
Strengths	Faster and easier to implement, as the same key is used to encrypt and decrypt data Requires less processing power Can be implemented in application-specific integrated chip (ASIC).	Convenient to use, as the distribution of keys to encrypt messages is not required
	Prevents widespread message security compromise as different secret keys are used to communicate with different parties	Enhanced security, as one need not share or transmit private keys to anyone
	The key is not bound to the data being transferred on the link; therefore, even if the data are intercepted, it is not possible to decrypt it	Provides digital signatures that cannot be repudiated
Weaknesses	**Symmetric Encryption**	**Asymmetric Encryption**
	Lack of secure channel to exchange the secret key	Slow in processing and requires high processing power

	Difficult to manage and secure too many shared keys that are generated to communicate with different parties	Widespread message security compromise is possible (i.e., an attacker can read complete messages if the private key is compromised)
	Provides no assurance about the origin and authenticity of a message, as the same key is used by both the sender and the receiver	Messages received cannot be decrypted if the private key is lost
	Vulnerable to dictionary attacks and brute-force attacks	Vulnerable to man-in-the-middle and brute-force attacks

Table 20.1: Strengths and weaknesses of crypto methods



Government Access to Keys (GAK)

- GAK means that software companies will give **copies of all keys** (or at least a sufficient proportion of each key that the remainder could be cracked) to the government
- The government promises that they will hold on to the keys in a **secure manner** and will only use them when a **court issues a warrant** to do so
- To the government, this is similar to the **ability to wiretapping phones**

Cryptographic Key

Item A

Item B

Items to which the GAK has right of access

Item C

Item D

Item E

• • •

Items to which the GAK has NO right of access

Government Access to Keys (GAK)

Government Access to Keys (GAK) refers to the statutory obligation of individuals and organizations to disclose their cryptographic keys to government agencies. It means that software companies will give copies of all keys (or at least enough of the key such that the remainder can be cracked) to the government. Law enforcement agencies around the world acquire and use these cryptographic keys to monitor suspicious communication and collect evidence of cybercrimes in the interests of national security. The government promises that it will hold on to the keys in a secure manner and only use them when a court issues a warrant to do so. To the government, this issue is similar to the ability to wiretap phones.

Government agencies often use key escrow for uninterrupted access to keys. Key escrow is a key exchange arrangement in which essential cryptographic keys are stored with a third party in escrow. The third party can use or allow others to use the encryption keys under certain predefined circumstances. The third party, with regard to GAK, is generally a government agency that may use the encryption keys to decipher digital evidence under authorization or a warrant from a court of law. However, there is growing concern about the privacy and security of cryptographic keys and information. Government agencies are responsible for protecting these keys. Such agencies generally use a single key to protect other keys, which is not a good idea, as revealing a single key could expose the other keys.

These agencies are not aware of how confidential the information protected by the keys is, which makes it difficult to judge how much protection is required. In cases where seized keys also protect other information that these agencies have no right to access, the consequences of key revelation cannot be determined, because government agencies are not aware of the information that the keys protect. In such cases, the key owner is liable for the consequences of key revelation. Before owners hand over their keys to government agencies, they need to be assured that the government agencies will protect these keys according to a sufficiently strong standard to protect their interests.

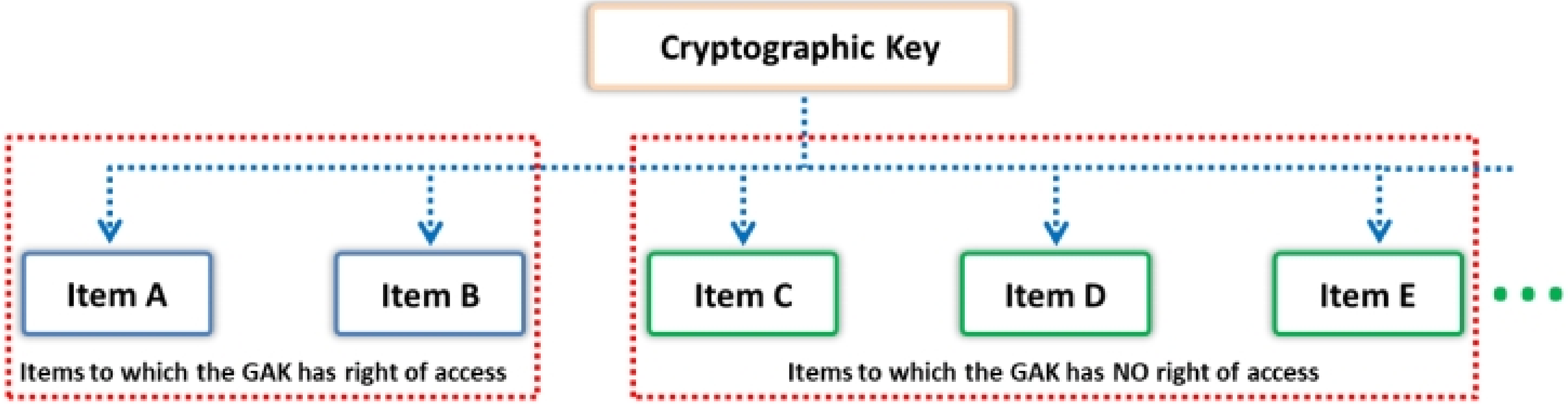
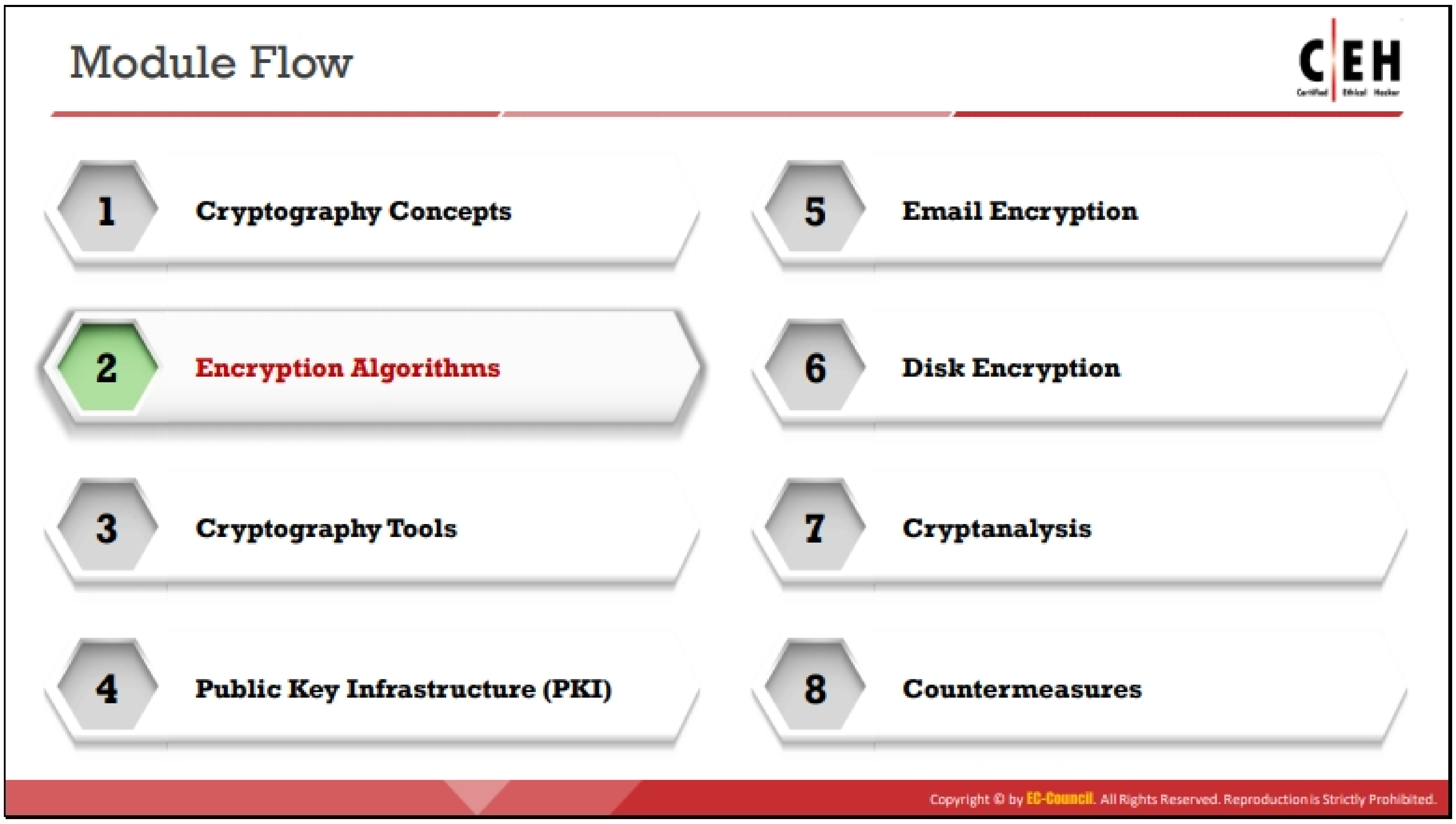

Cryptographic Key
Item A
Item B
Item C
Item D
Item E
Items to which the GAK has right of access
Items to which the GAK has NO right of access

Figure 20.4: Illustration of GAK



Module Flow

1 Cryptography Concepts

2 Encryption Algorithms

3 Cryptography Tools

4 Public Key Infrastructure (PKI)

5 Email Encryption

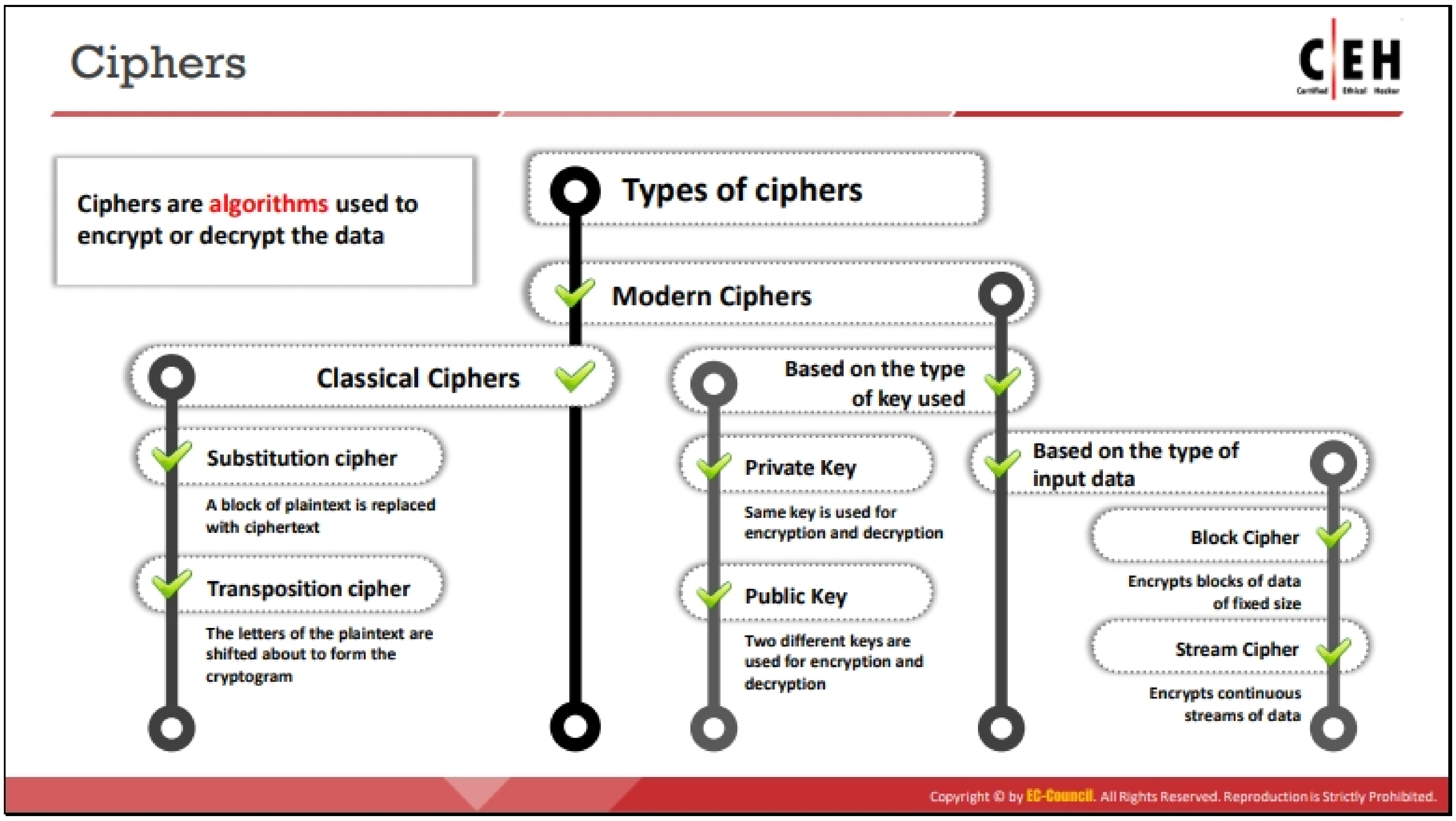
6 Disk Encryption

7 Cryptanalysis

8 Countermeasures

Encryption Algorithms

Encryption is the process of converting readable plaintext into an unreadable ciphertext using a set of complex algorithms that transform the data into blocks or streams of random alphanumeric characters. This section deals with ciphers and various encryption algorithms such as DES, AES, RC4, RC5, RC6, DSA, RSA, MD5, SHA, etc.



Ciphers

Ciphers are algorithms used to encrypt or decrypt the data

Types of ciphers

Modern Ciphers

Classical Ciphers

Substitution cipher

A block of plaintext is replaced with ciphertext

Transposition cipher

The letters of the plaintext are shifted about to form the cryptogram

Based on the type of key used

Private Key

Same key is used for encryption and decryption

Public Key

Two different keys are used for encryption and decryption

Based on the type of input data

Block Cipher

Encrypts blocks of data of fixed size

Stream Cipher

Encrypts continuous streams of data

Ciphers

In cryptography, a cipher is an algorithm (a series of well-defined steps) for performing encryption and decryption. Encipherment is the process of converting plaintext into a cipher or code; the reverse process is called decipherment. A message encrypted using a cipher is rendered unreadable unless its recipient knows the secret key required to decrypt it. Communication technologies (e.g., Internet, cell phones) rely on ciphers to maintain both security and privacy. Cipher algorithms may be open-source (the algorithmic process is in the public domain while the key is selected by a user and is private) or closed-source (the process is developed for use in specific domains, such as the military, and the algorithm itself is not in the public domain). Furthermore, ciphers may be free for public use or licensed.

Types of ciphers

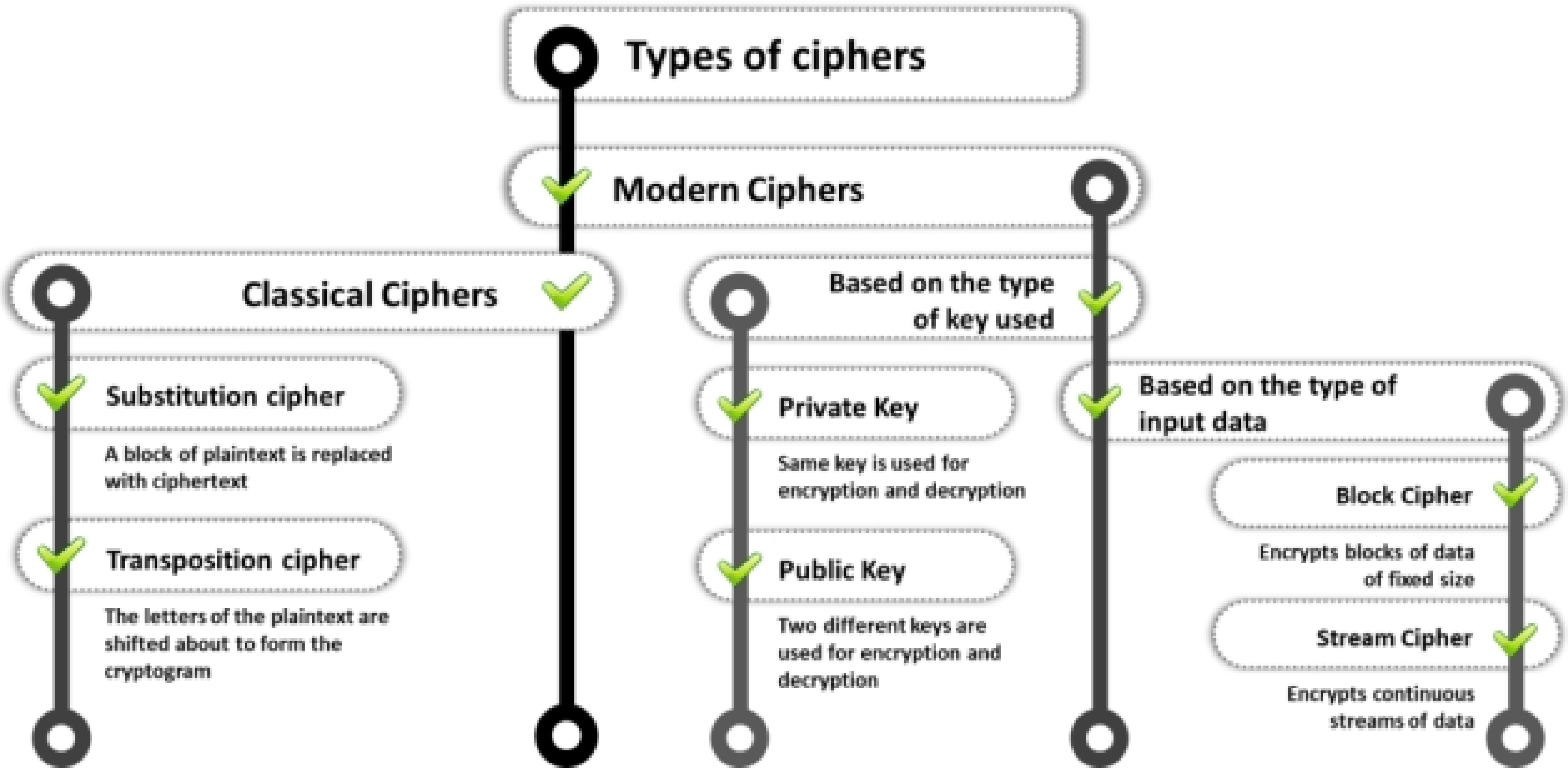

Types of ciphers
Modern Ciphers
Classical Ciphers
Substitution cipher
A block of plaintext is replaced with ciphertext
Transposition cipher
The letters of the plaintext are shifted about to form the cryptogram
Based on the type of key used
Private Key
Same key is used for encryption and decryption
Public Key
Two different keys are used for encryption and decryption
Based on the type of input data
Block Cipher
Encrypts blocks of data of fixed size
Stream Cipher
Encrypts continuous streams of data

Figure 20.5: Classification of ciphers

Ciphers are of two main types: classical and modern.

- **Classical Ciphers**

  Classical ciphers are the most basic type of ciphers, which operate on letters of the alphabet (A–Z). These ciphers are generally implemented either by hand or with simple mechanical devices. Because these ciphers are easily deciphered, they are generally unreliable.

  Types of classical ciphers

  - **Substitution cipher**: The user replaces units of plaintext with ciphertext according to a regular system. The units may be single letters, pairs of letters, or combinations of them, and so on. The recipient performs inverse substitution to decipher the text. Examples include the Beale cipher, autokey cipher, Gronsfeld cipher, and Hill cipher.

    For example, "**HELLO WORLD**" can be encrypted as "**PSTER HGFST**" (i.e., H=P, E=S, etc.).

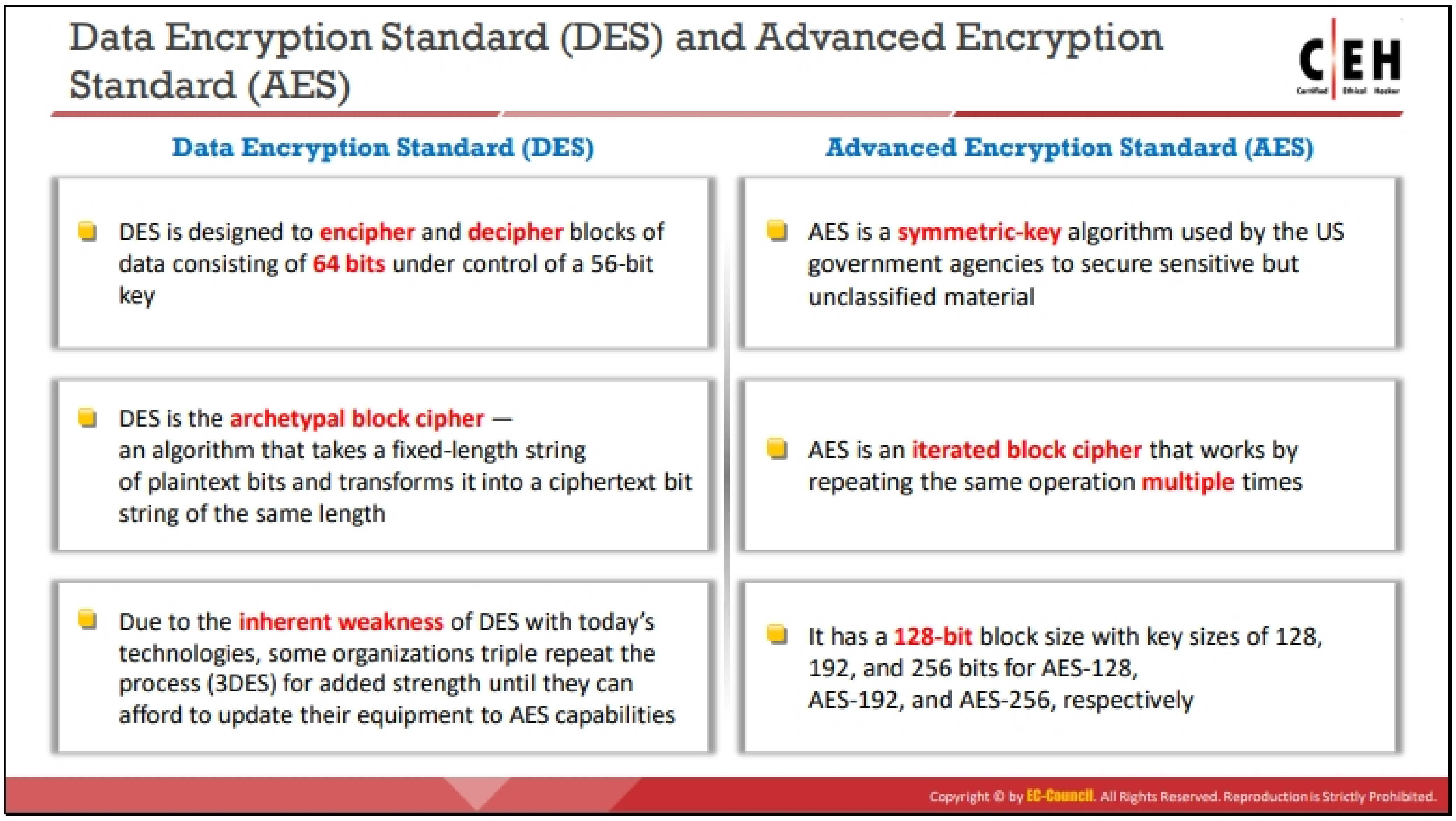
  - **Transposition cipher**: Here, letters in the plaintext are rearranged according to a regular system to produce the ciphertext. For example, "**CRYPTOGRAPHY**" when encrypted becomes "**AOYCRGPTYRHP**." Examples include the rail fence cipher, route cipher, and Myszkowski transposition.

- **Modern Ciphers**

  Modern ciphers are designed to withstand a wide range of attacks. They provide message secrecy, integrity, and authentication of the sender. A user can calculate a modern cipher using a one-way mathematical function that is capable of factoring large prime numbers.

  Types of Modern ciphers

  - **Based on the type of key used**
    - **Symmetric-key algorithms (Private-key cryptography)**: Use the same key for encryption and decryption.
    - **Asymmetric-key algorithms (Public-key cryptography)**: Use two different keys for encryption and decryption.
  - **Based on the type of input data**
    - **Block cipher**: Deterministic algorithms operating on a block (a group of bits) of fixed size with an unvarying transformation specified by a symmetric key. Most modern ciphers are block ciphers. They are widely used to encrypt bulk data. Examples include DES, AES, IDEA, etc. When the block size is less than that used by the cipher, padding is employed to achieve a fixed block size.
    - **Stream cipher**: Symmetric-key ciphers are plaintext digits combined with a key stream (pseudorandom cipher digit stream). Here, the user applies the key to each bit, one at a time. Examples include RC4, SEAL, etc.



Data Encryption Standard (DES) and Advanced Encryption Standard (AES)

Data Encryption Standard (DES)

- DES is designed to **encipher** and **decipher** blocks of data consisting of **64 bits** under control of a 56-bit key
- DES is the **archetypal block cipher** — an algorithm that takes a fixed-length string of plaintext bits and transforms it into a ciphertext bit string of the same length
- Due to the **inherent weakness** of DES with today's technologies, some organizations triple repeat the process (3DES) for added strength until they can afford to update their equipment to AES capabilities

Advanced Encryption Standard (AES)

- AES is a **symmetric-key** algorithm used by the US government agencies to secure sensitive but unclassified material
- AES is an **iterated block cipher** that works by repeating the same operation **multiple** times
- It has a **128-bit** block size with key sizes of 128, 192, and 256 bits for AES-128, AES-192, and AES-256, respectively

Data Encryption Standard (DES)

DES is a standard for data encryption that uses a secret key for both encryption and decryption (symmetric cryptosystem). DES uses a 64-bit secret key, of which 56 bits are generated randomly and the other 8 bits are used for error detection. It uses a data encryption algorithm (DEA), a secret key block cipher employing a 56-bit key operating on 64-bit blocks. DES is the archetypal block cipher—an algorithm that takes a fixed-length string of plaintext bits and transforms it into a ciphertext bit string of the same length. The design of DES allows users to implement it in hardware and use it for single-user encryption, such as to store files on a hard disk in encrypted form.

DES provides 72 quadrillion or more possible encryption keys and chooses a random key for the encryption of each message. Because of the inherent weakness of DES vis-à-vis today's technologies, some organizations use triple DES (3DES), in which they repeat the process three times for added strength until they can afford to update their equipment to AES capabilities.

Triple Data Encryption Standard (3DES)

Eventually, it became obvious that DES would no longer be secure. The U.S. Federal Government began a contest seeking a replacement cryptography algorithm. However, in the meantime, 3DES was created as an interim solution. Essentially, it performs DES three times with three different keys. 3DES uses a "key bundle" that comprises three DES keys, K1, K2, and K3. Each key is a standard 56-bit DES key. It then performs the following process:

DES *encrypt* with K1, DES *decrypt* with K2, DES *encrypt* with K3

There are three options for the keys. In the first option, all three keys are independent and different. In the second option, K1 and K3 are identical. In the third option, all three keys are the same; therefore, you are literally applying the same DES algorithm three times with the same key. The first option is the most secure, while the third is the least secure.

Advanced Encryption Standard (AES)

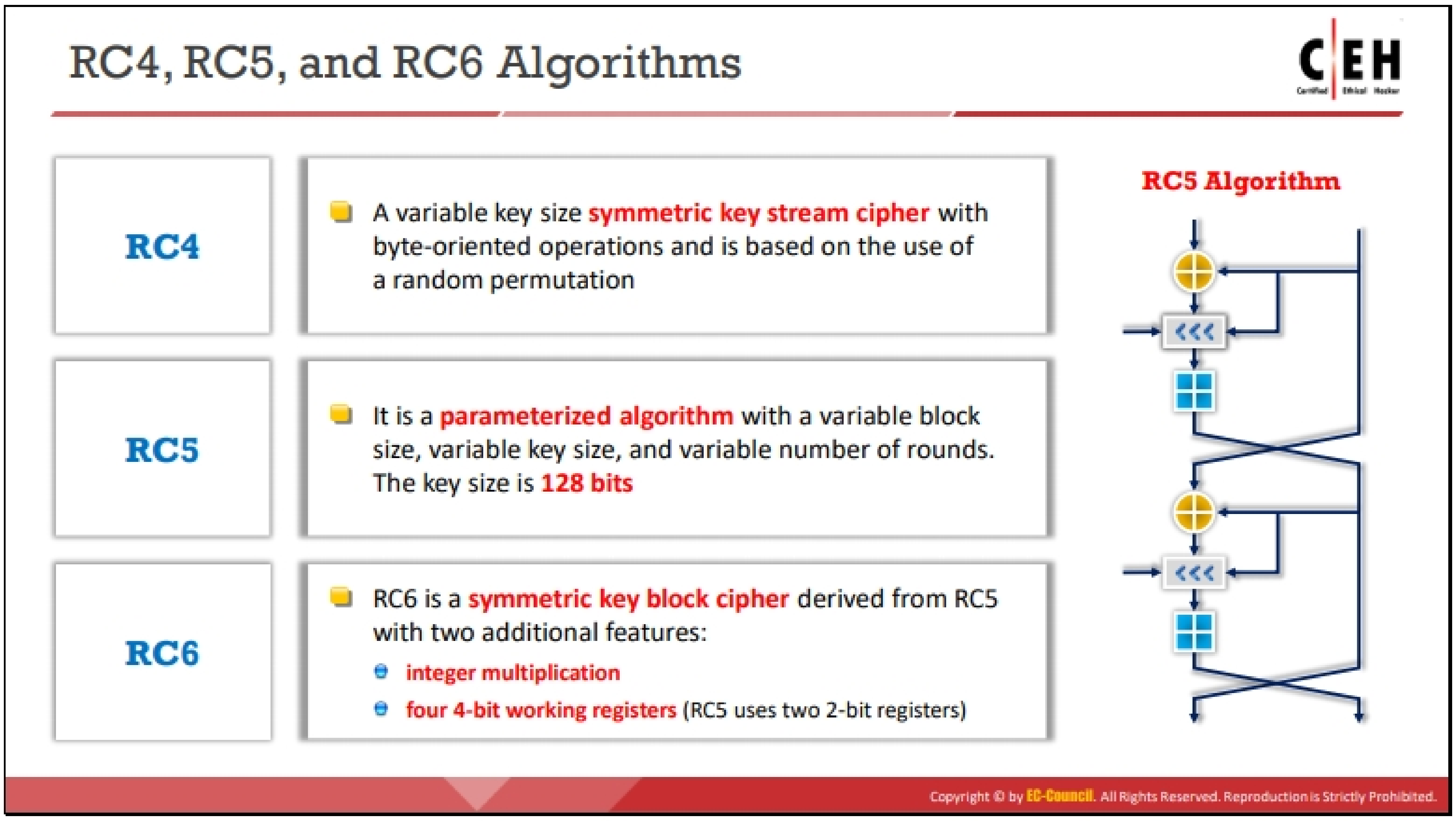
The Advanced Encryption Standard (AES) is a National Institute of Standards and Technology (NIST) specification for the encryption of electronic data. It also helps to encrypt digital information such as telecommunications, financial, and government data. US government agencies have been using it to secure sensitive but unclassified material.

AES consists of a symmetric-key algorithm: both encryption and decryption are performed using the same key. It is an iterated block cipher that works by repeating the defined steps multiple times. It has a 128-bit block size, with key sizes of 128, 192, and 256 bits for AES-128, AES-192, and AES-256, respectively. The design of AES makes its use efficient in both software and hardware. It works simultaneously at multiple network layers.

AES Pseudocode

Initially, the system copies the cipher input into the internal state and then adds an initial round key. The system transforms the state by iterating a round function in a number of cycles. The number of cycles may vary with the block size and key length. After completing rounding, the system copies the final state into the cipher output.

```
Cipher (byte in [4*Nb], byte out [4*Nb], word w[Nb*(Nr+1)])
begin
  byte state[4, Nb]
  state = in
  AddRoundKey (state, w)
    for round = 1 step 1 to Nr-1
       SubBytes(state)
       ShiftRows(state)
       MixColumns(state)
       AddRoundKey(state, w+round*Nb)
    end for
  SubBytes(state)
  ShiftRows(state)
  AddRoundKey(state, w+Nr*Nb)
  out = state
end
```



RC4, RC5, and RC6 Algorithms

RC4	A variable key size **symmetric key stream cipher** with byte-oriented operations and is based on the use of a random permutation
RC5	It is a **parameterized algorithm** with a variable block size, variable key size, and variable number of rounds. The key size is **128 bits**
RC6	RC6 is a **symmetric key block cipher** derived from RC5 with two additional features: **integer multiplication** **four 4-bit working registers** (RC5 uses two 2-bit registers)

RC5 Algorithm

RC4, RC5, and RC6 Algorithms

Symmetric encryption algorithms developed by RSA Security are discussed below.

- **RC4**

  RC4 is a variable key-size symmetric-key stream cipher with byte-oriented operations, and it is based on the use of a random permutation. According to some analyses, the period of the cipher is likely to be greater than 10,100. Each output byte uses 8 to 16 system operations; thus, the cipher can run fast when used in software. RC4 enables safe communications such as for traffic encryption (which secures websites) and for websites that use the SSL protocol.

- **RC5**

  RC5 is a fast symmetric-key block cipher designed by Ronald Rivest for RSA Data Security (now RSA Security). The algorithm is a parameterized algorithm with a variable block size, a variable key size, and a variable number of rounds. The block sizes can be 32, 64, or 128 bits. The range of the rounds can vary from 0 to 255, and the size of the key can vary from 0 to 2,040 bits. This built-in variability can offer flexibility at all levels of security. The routines used in RC5 are key expansion, encryption, and decryption.

  In the key expansion routine, the secret key that a user provides is expanded to fill the key table (the size of which depends on the number of rounds). RC5 uses a key table for both encryption and decryption. The encryption routine has three fundamental operations: integer addition, bitwise XOR, and variable rotation. The intensive use of data-dependent rotation and the combination of different operations make RC5 a secure encryption algorithm.

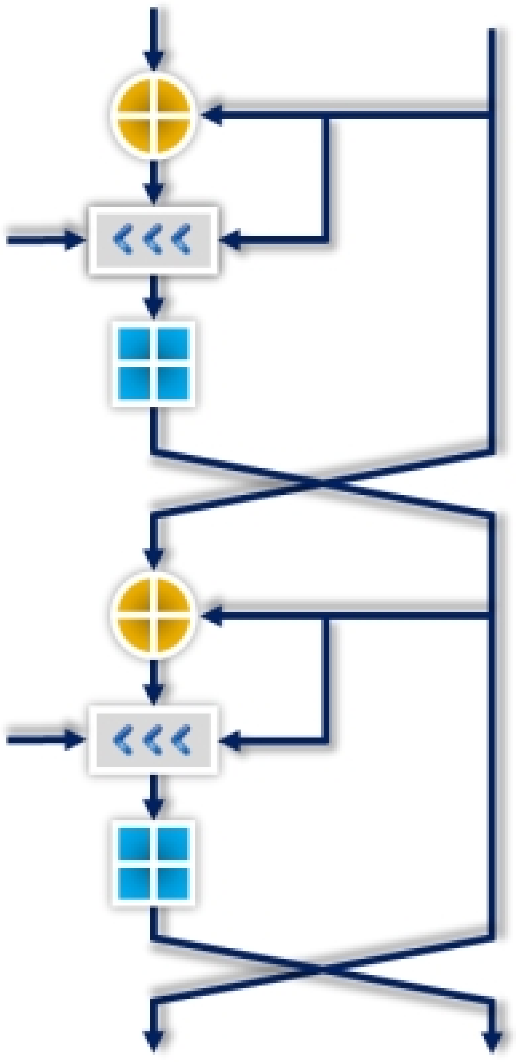


Figure 20.6: Block diagram of the RC5 algorithm

- **RC6**

  RC6 is a symmetric-key block cipher derived from RC5. It is a parameterized algorithm with a variable block size, key size, and number of rounds. Two features that differentiate RC6 from RC5 are integer multiplication (which is used to increase the diffusion, achieved in fewer rounds with increased speed of the cipher) and the use of four 4-bit working registers rather than two 2-bit registers. RC6 uses four 4-bit registers instead of two 2-bit registers because the block size of the AES is 128 bits.

Blowfish

Blowfish is a type of symmetric block cipher algorithm designed to replace DES or IDEA algorithms. It uses the same secret key to encrypt and decrypt data. This algorithm splits the data into a block length of 64 bits and produces a key ranging from 32 bits to 448 bits. Due to its high speed and overall efficiency, blowfish is used in software ranging from password protection tools to e-commerce websites for securing payments.

It is a 16-round Feistel cipher working on 64-bit blocks. However, unlike DES, its key size ranges from 32 bits to 448 bits.

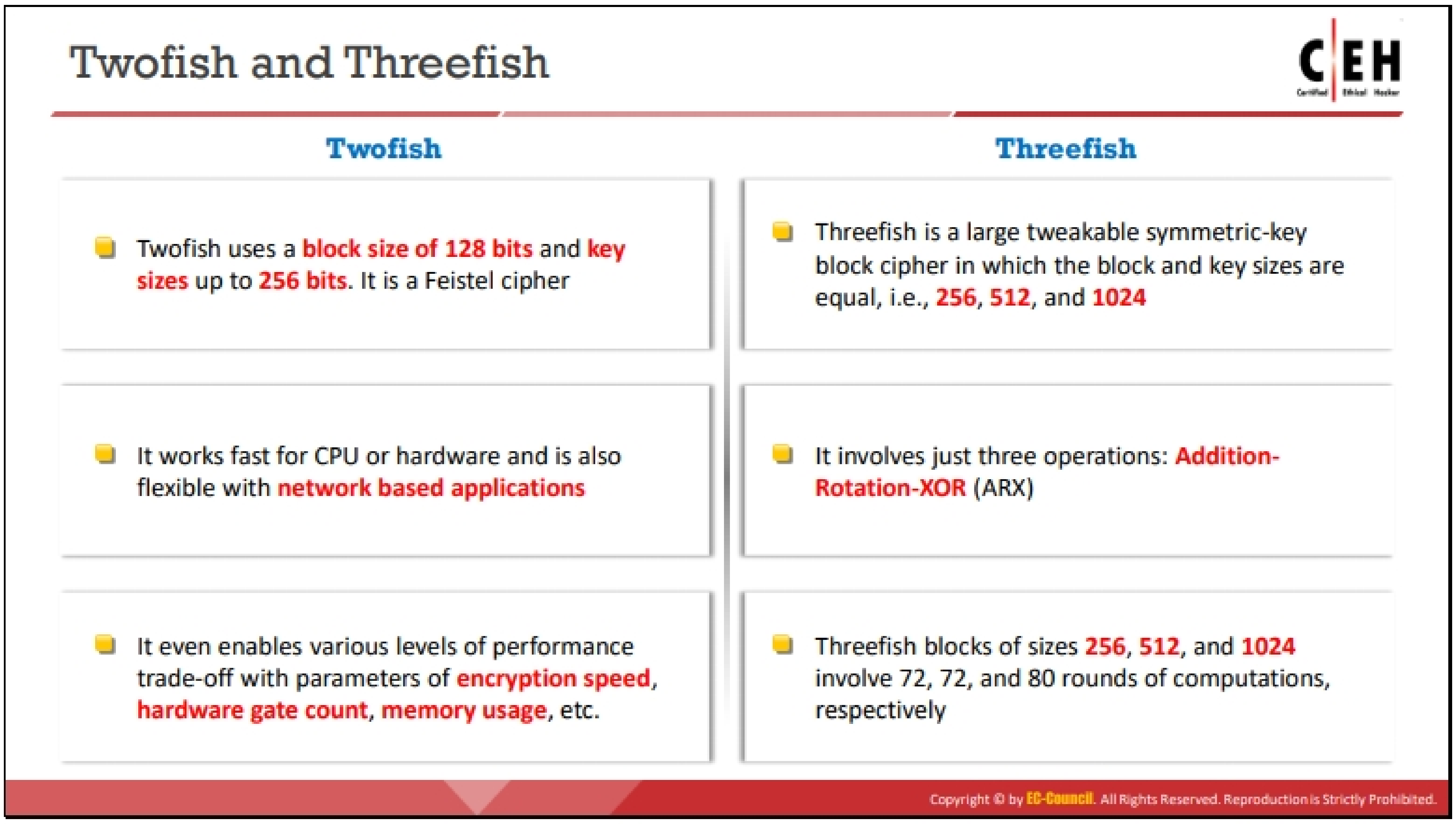
This algorithm has two parts. The first part handles the expansion of the key. The second part actually encrypts the data.

The key expansion is handled in several steps. The first step is to break the original key into a set of subkeys. Specifically, a key of no more than 448 bits is separated into 4,168 bytes. There is a P-array and four 32-bit S-boxes. The P-array contains 18 32-bit subkeys, while each S-box contains 256 entries.

Key expansion is performed as follows:

1. The first step is to initialize the **P-array** and **S-boxes**.
2. Then, XOR the P-array with the key bits. For example, **P1 XOR** (first 32 bits of the key), **P2 XOR** (next 32 bits of the key).
3. Use the above method to encrypt the **all-zero string**.
4. This new output is now P1 and P2.
5. Encrypt the new P1 and P2 with the **modified subkeys**.
6. This new output is now P3 and P4.
7. Repeat the process **521 times** to calculate new subkeys for the P-array and the four S-boxes.

The round function splits the 32-bit input into four 8-bit quarters and uses the quarters as input to the S-boxes. The outputs are added modulo **232** and **XORed** to produce the final **32-bit output**.



Twofish and Threefish

Twofish

- Twofish uses a **block size of 128 bits** and **key sizes** up to **256 bits**. It is a Feistel cipher
- It works fast for CPU or hardware and is also flexible with **network based applications**
- It even enables various levels of performance trade-off with parameters of **encryption speed**, **hardware gate count**, **memory usage**, etc.

Threefish

- Threefish is a large tweakable symmetric-key block cipher in which the block and key sizes are equal, i.e., **256**, **512**, and **1024**
- It involves just three operations: **Addition-Rotation-XOR** (ARX)
- Threefish blocks of sizes **256**, **512**, and **1024** involve 72, 72, and 80 rounds of computations, respectively

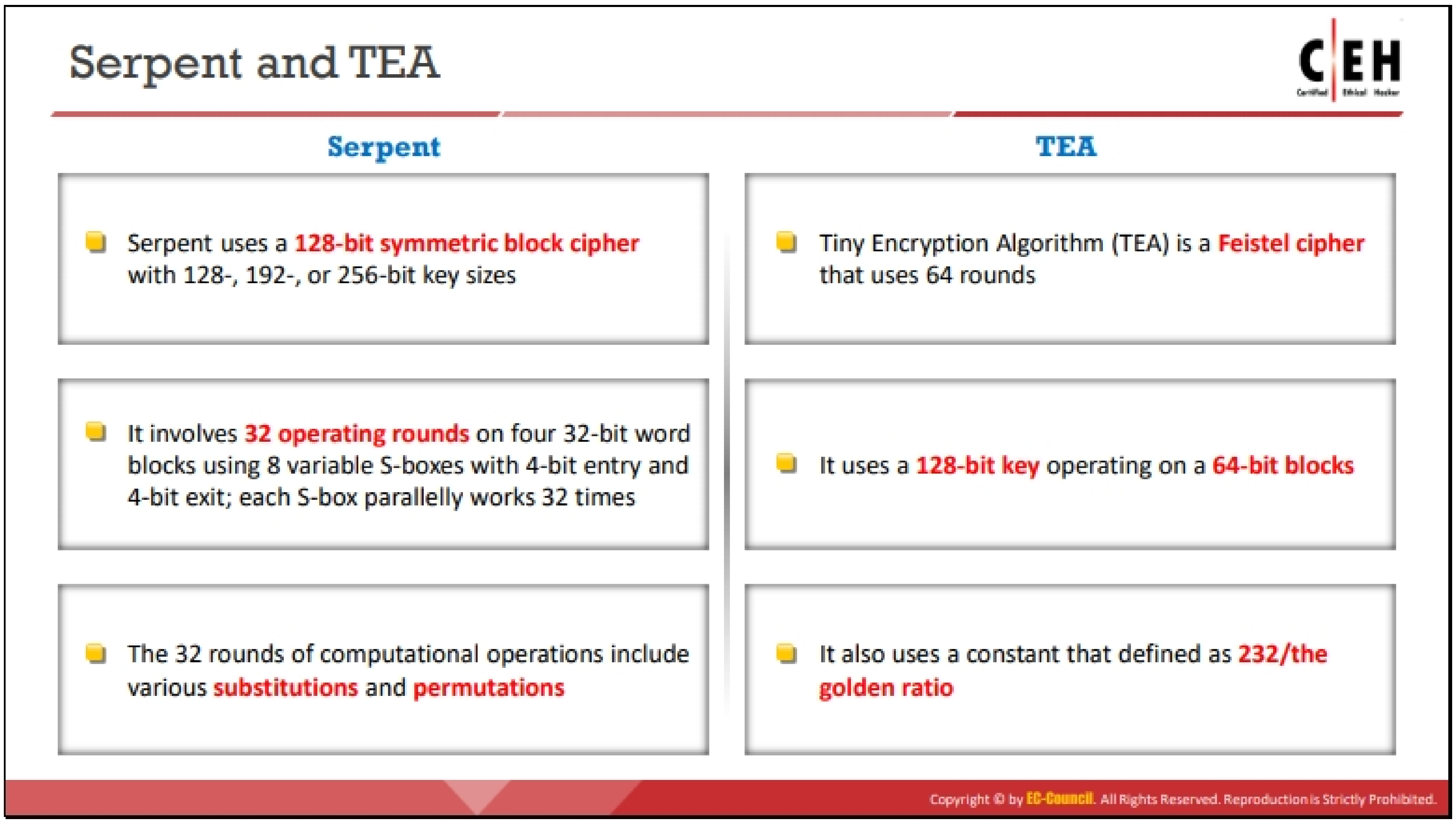
Twofish

The Twofish algorithm was one of the U.S. Government's five finalists to replace DES, but it was not chosen. It was designed by Bruce Schneier, John Kelsey, Doug Whiting, David Wagner, Chris Hall, and Niels Ferguson.

TwoFish is a 128-bit block cipher. It is one of the most conceptually simple algorithms that uses a single key for both encryption and decryption for any length up to 256 bits. It is a Feistel cipher. It not only works fast for CPU or hardware but is also flexible for network-based applications. Furthermore, it allows various levels of performance trade-off on parameters such as encryption speed, hardware gate count, memory usage, etc. This technique of enabling different implementations improves the relative performance of the algorithm. Any user can optimize the performance based on the key scheduling.

Threefish

Threefish was developed in 2008 and it is a part of the Skein algorithm. It was enrolled in NIST's SHA-3 (hash function) contest. It is a large tweakable symmetric-key block cipher in which the block and key sizes are equal, i.e., 256, 512, and 1024. Threefish involves only three operations, i.e., ARX (addition-rotation-XOR), which makes the coding simple, and all these operations work on 64-bit words. Threefish blocks 256, 512, and 1024 involve 72, 72, and 80 rounds of computations, respectively, to achieve the final security goal. This algorithm does not use S-boxes to prevent cache timing attacks.



Serpent and TEA

Serpent

- Serpent uses a **128-bit symmetric block cipher** with 128-, 192-, or 256-bit key sizes
- It involves **32 operating rounds** on four 32-bit word blocks using 8 variable S-boxes with 4-bit entry and 4-bit exit; each S-box parallelly works 32 times
- The 32 rounds of computational operations include various **substitutions** and **permutations**

TEA

- Tiny Encryption Algorithm (TEA) is a **Feistel cipher** that uses 64 rounds
- It uses a **128-bit key** operating on a **64-bit blocks**
- It also uses a constant that defined as **232/the golden ratio**

Serpent

Like Blowfish, Serpent is a symmetric-key block cipher that was a finalist in the AES contest. This algorithm was designed by Ross Anderson, Eli Biham, and Lars Knudsen. It uses a 128-bit symmetric block cipher with key sizes of 128, 192, or 256 bits. It can be integrated into software or hardware programs without any restrictions.

Serpent involves 32 rounds of computational operations that include substitution and permutation operations on four 32-bit word blocks using 8-variable S-boxes with 4-bit entry and 4-bit exit. All S-boxes work parallelly 32 times. Although, Serpent is one of the most secure encryption mechanisms in AES contests, researchers have chosen Rijndael over Serpent due to its moderate encryption speed (owing to the number of rounds it uses) and complexity. Serpent minimizes the correlation between encoded images or plaintexts to a greater extent compared to Twofish and Rijndael. Therefore, Rijndael is the stand-out AES competitor and is now being used as AES.

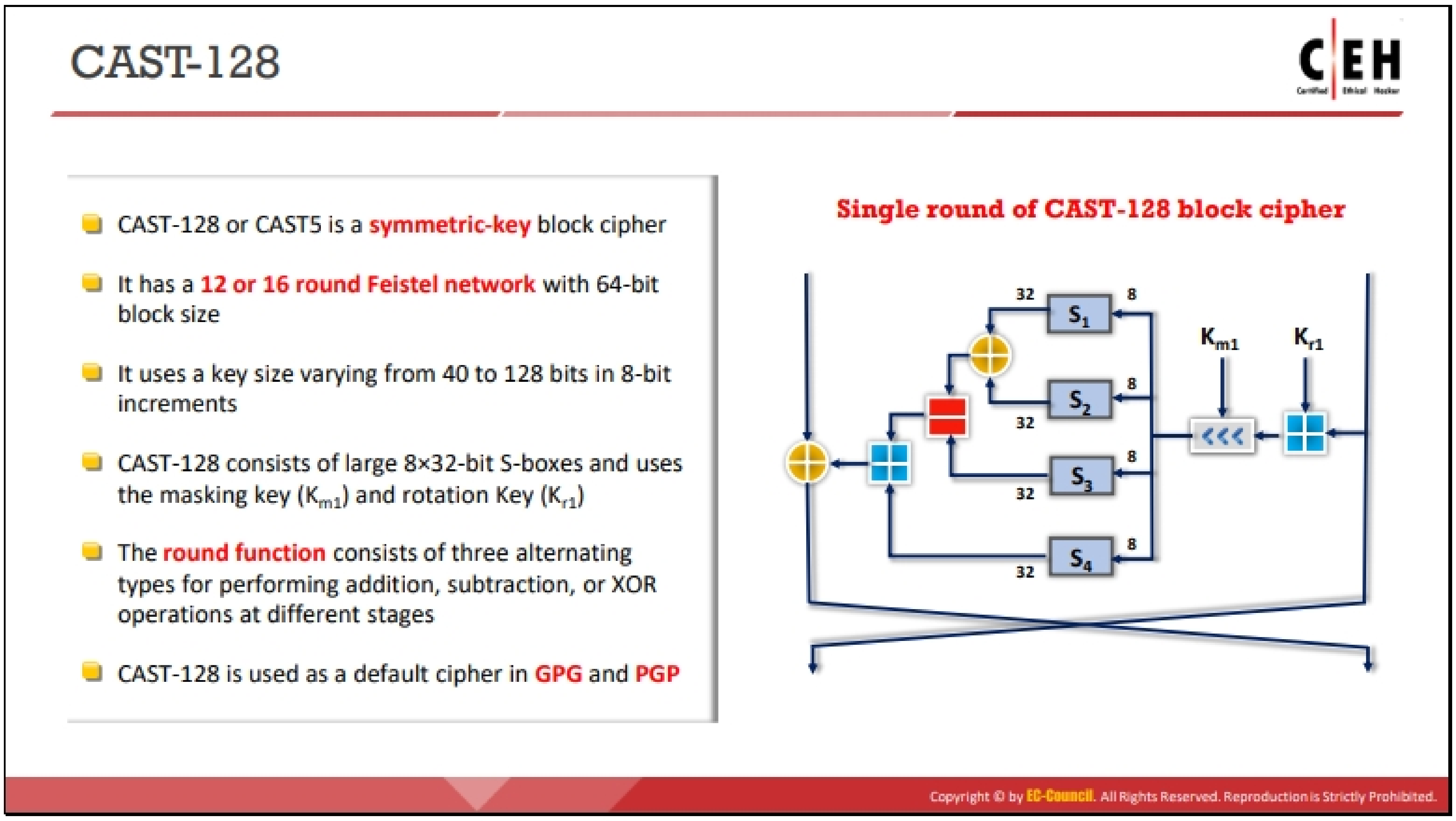
TEA

The tiny encryption algorithm (TEA) was created by David Wheeler and Roger Needham, and it was publicly presented for the first time in 1994. It is a simple algorithm, easy to implement in code. It is a Feistel cipher that uses 64 rounds (note that this is a suggestion; it can be implemented with fewer or more rounds). The number of rounds should be even since they are implemented in pairs called cycles.

TEA uses a 128-bit key operating on a 64-bit block. It also uses a constant that is defined as 232/the golden ratio. This constant is referred to as delta, and in each round, a multiple of delta is used. The 128-bit key is split into four different 32-bit subkeys labeled K[0], K[1], K[2], and K[3].

Instead of using the XOR operation, TEA uses addition and subtraction, but with mod 232. The block is divided into two halves, R and L. R is processed through the round function.

The round function takes the R half and performs a left shift of 4. Then, the result of this operation is added to K[0]. Next, the result of this operation is added to delta (recall that delta is the current multiple of 232/the golden ratio). The result of this operation is then shifted right by 5 and added to K[1]. This is the round function. As with all Feistel ciphers, the result of the round function is XORed with L, and L and R are then swapped for the next round.



CAST-128

- CAST-128 or CAST5 is a **symmetric-key** block cipher
- It has a **12 or 16 round Feistel network** with 64-bit block size
- It uses a key size varying from 40 to 128 bits in 8-bit increments
- CAST-128 consists of large 8×32-bit S-boxes and uses the masking key (K_{m1}) and rotation Key (K_{r1})
- The **round function** consists of three alternating types for performing addition, subtraction, or XOR operations at different stages
- CAST-128 is used as a default cipher in **GPG** and **PGP**

Single round of CAST-128 block cipher

CAST-128

CAST-128, also called CAST5, is a symmetric-key block cipher having a classical 12- or 16-round Feistel network with a block size of 64 bits. CAST-128 uses a key size varying from 40 bits to 128 bits in 8-bit increments. The CAST-128 components include large 8×32-bit S-boxes (S1, S2, S3, S4) based on bent functions, modular addition and subtraction, key-dependent rotation, and XOR operations. CAST-128 uses a masking key (K_{m1}) and a rotation key (K_{r1}) for performing its functions. The round function consists of three alternating types to perform addition, subtraction, or XOR operations in different stages. It used as a default cipher in GPG (GNU Privacy Guard) and PGP (Pretty Good Privacy).

CAST-256 is an extension of CAST-128 that uses the same design procedure. CAST-256 has a 128-bit block size, and it uses key sizes varying from 128 to 256 bits. Furthermore, it uses zero-correlation cryptanalysis, which can break 28 rounds with time = 2246.9 and data = 298.8.

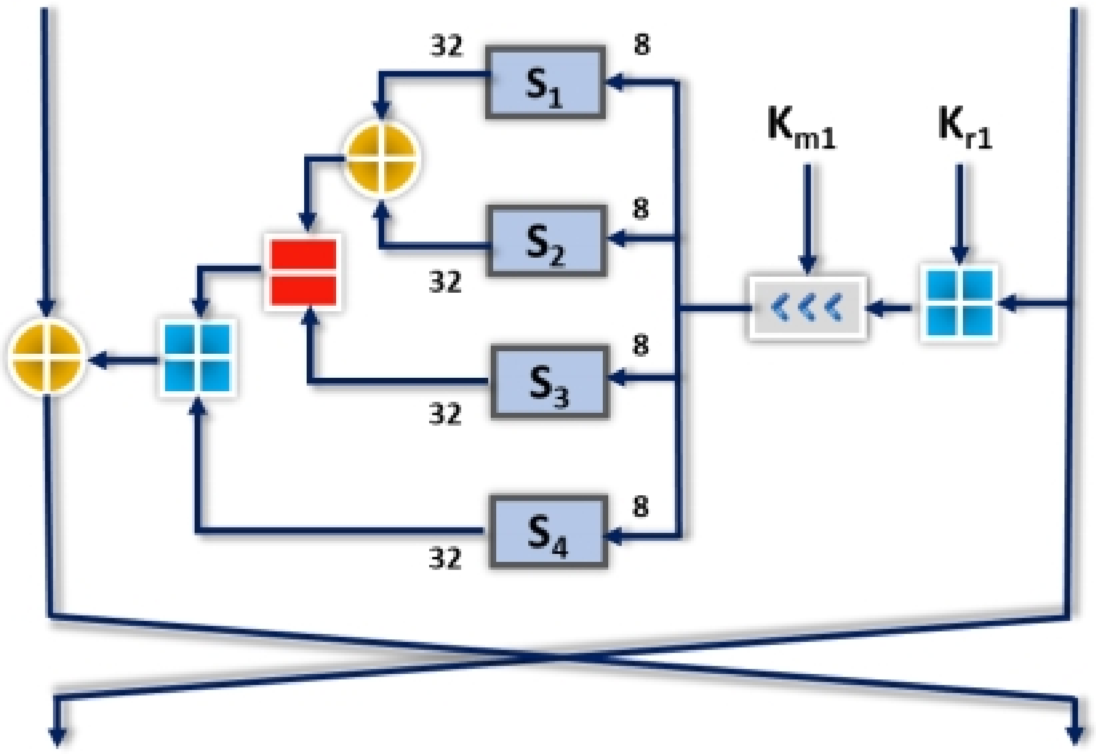
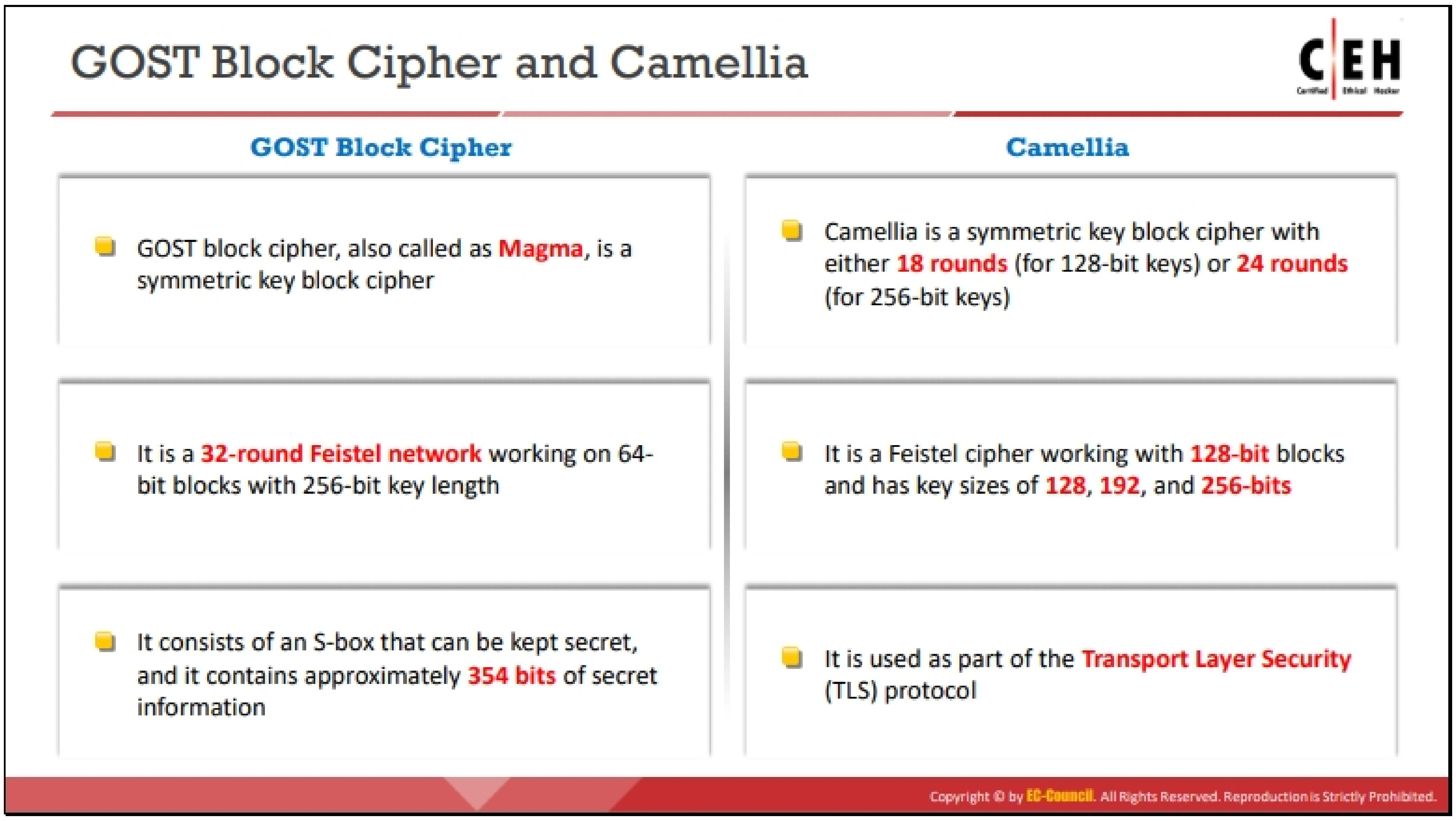


Figure 20.7: Block diagram of CAST-128



GOST Block Cipher and Camellia

GOST Block Cipher

- GOST block cipher, also called as **Magma**, is a symmetric key block cipher
- It is a **32-round Feistel network** working on 64-bit blocks with 256-bit key length
- It consists of an S-box that can be kept secret, and it contains approximately **354 bits** of secret information

Camellia

- Camellia is a symmetric key block cipher with either **18 rounds** (for 128-bit keys) or **24 rounds** (for 256-bit keys)
- It is a Feistel cipher working with **128-bit** blocks and has key sizes of **128**, **192**, and **256-bits**
- It is used as part of the **Transport Layer Security** (TLS) protocol

GOST Block Cipher

The GOST (Government Standard) block cipher, also called Magma, is a symmetric-key block cipher having a 32-round Feistel network working on 64-bit blocks with a 256-bit key length. It consists of an S-box that can be kept secret and it contains around 354 bits of secret information. GOST is a simple encryption algorithm, where the round function 32-bit subkey modulo 232 is added and put in the layer of S-boxes and the rotate left shift operation is used for shifting 11 bits, thereby providing the output of the round function.

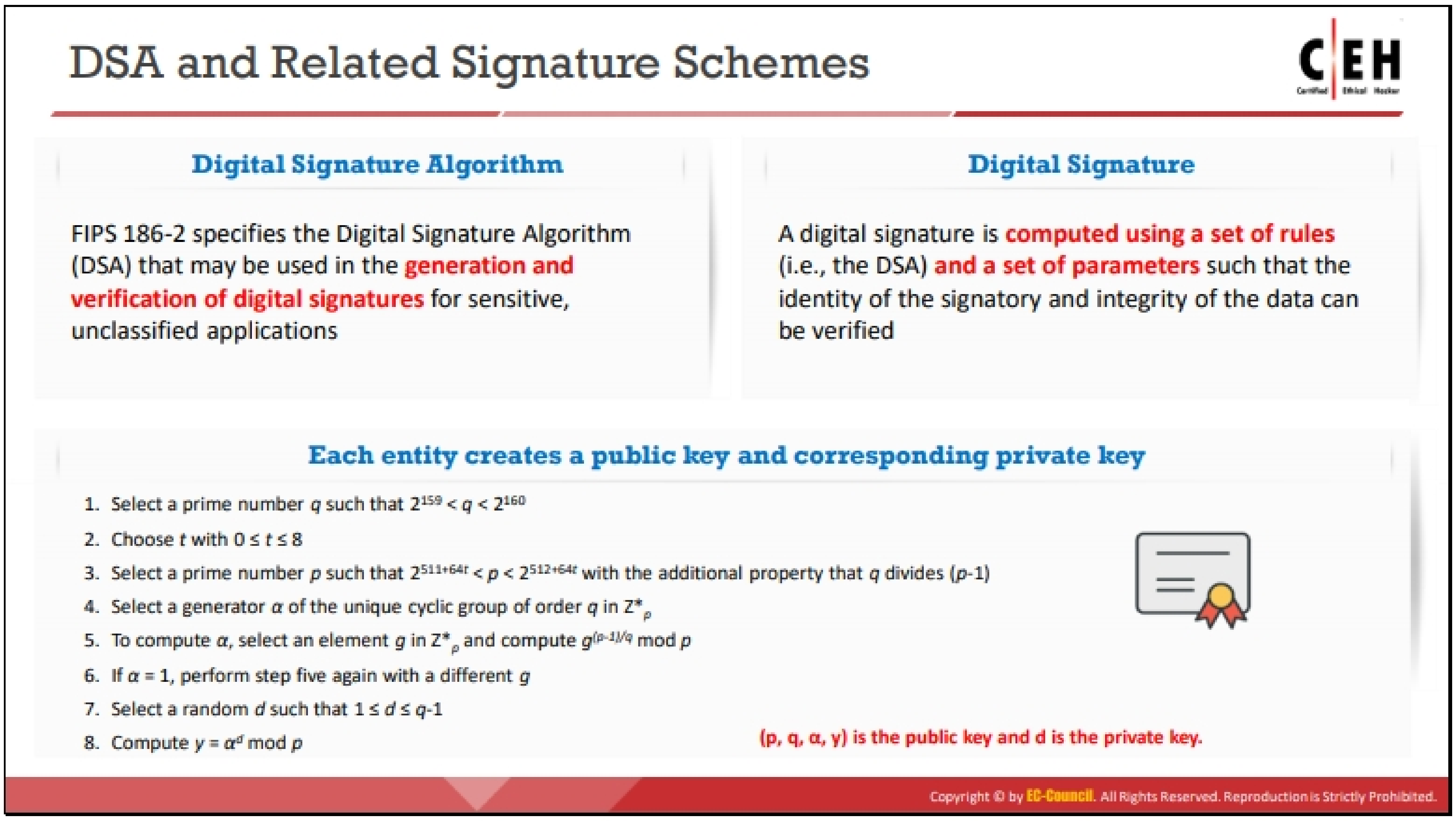
The key scheduling of the GOST block cipher is performed by breaking the 256-bit key into eight 32-bit subkeys, where each subkey is used four times. In this algorithm, the key words are used in order for the first 24 rounds and they are used in reverse order for the last 8 rounds.

Kuznyechik is the latest extension of GOST, which uses 128-bit blocks.

Camellia

Camellia is a symmetric-key block cipher having either 18 rounds (for 128-bit keys) or 24 rounds (for 256-bit keys). It is a Feistel cipher with a block size of 128 bits and a key size of 128, 192, and 256 bits. Camellia uses four 8×8-bit S-boxes that perform affine transformations and logical operations. A logical transformation layer FL-function or its inverse is applied every six rounds. Camellia uses the key whitening technique for increased security.

Camellia is a part of the Transport Layer Security (TLS) protocol, which is used to deliver secure communication. Camellia cannot be brute-forced even with the latest technology although it uses a smaller key size of 128 bits, thus making it a safe cipher. In addition, Camellia offers high security and its processing skills are equivalent to those of AES or Rijndael.



DSA and Related Signature Schemes

Digital Signature Algorithm

FIPS 186-2 specifies the Digital Signature Algorithm (DSA) that may be used in the **generation and verification of digital signatures** for sensitive, unclassified applications

Digital Signature

A digital signature is **computed using a set of rules** (i.e., the DSA) **and a set of parameters** such that the identity of the signatory and integrity of the data can be verified

Each entity creates a public key and corresponding private key

1. Select a prime number q such that $2^{159} < q < 2^{160}$
2. Choose t with $0 \le t \le 8$
3. Select a prime number p such that $2^{511+64t} < p < 2^{512+64t}$ with the additional property that q divides $(p\text{-}1)$
4. Select a generator α of the unique cyclic group of order q in Z^*_p
5. To compute α, select an element g in Z^*_p and compute $g^{(p-1)/q}$ mod p
6. If $\alpha = 1$, perform step five again with a different g
7. Select a random d such that $1 \le d \le q\text{-}1$
8. Compute $y = \alpha^d$ mod p

(p, q, α, y) is the public key and d is the private key.

DSA and Related Signature Schemes

The Digital Signature Algorithm (DSA) is a Federal Information Processing Standard for digital signatures. The NIST proposed the DSA for use in the Digital Signature Standard (DSS), adopted as FIPS 186. The DSA helps in the generation and verification of digital signatures for sensitive and unclassified applications. It creates a 320-bit digital signature with 512–1024-bit security.

A digital signature is a mathematical scheme used for the authentication of digital messages. Computation of the digital signature uses a set of rules (i.e., the DSA) and a set of parameters in that the user can verify the identity of the signatory and the integrity of the data.

Processes involved in DSA:

- **Signature Generation Process**: The private key is used to know who has signed it.
- **Signature Verification Process**: The public key is used to verify whether the given digital signature is genuine.

DSA is a public-key cryptosystem, as it involves the use of both private and public keys.

Benefits of DSA:

- Less chances of forgery compared with a written signature
- Quick and easy method of business transactions
- Fake currency problem can be mitigated considerably

DSA Algorithm:

Each entity A does the following:

1. Select a prime number q such that $2^{159} < q < 2^{160}$

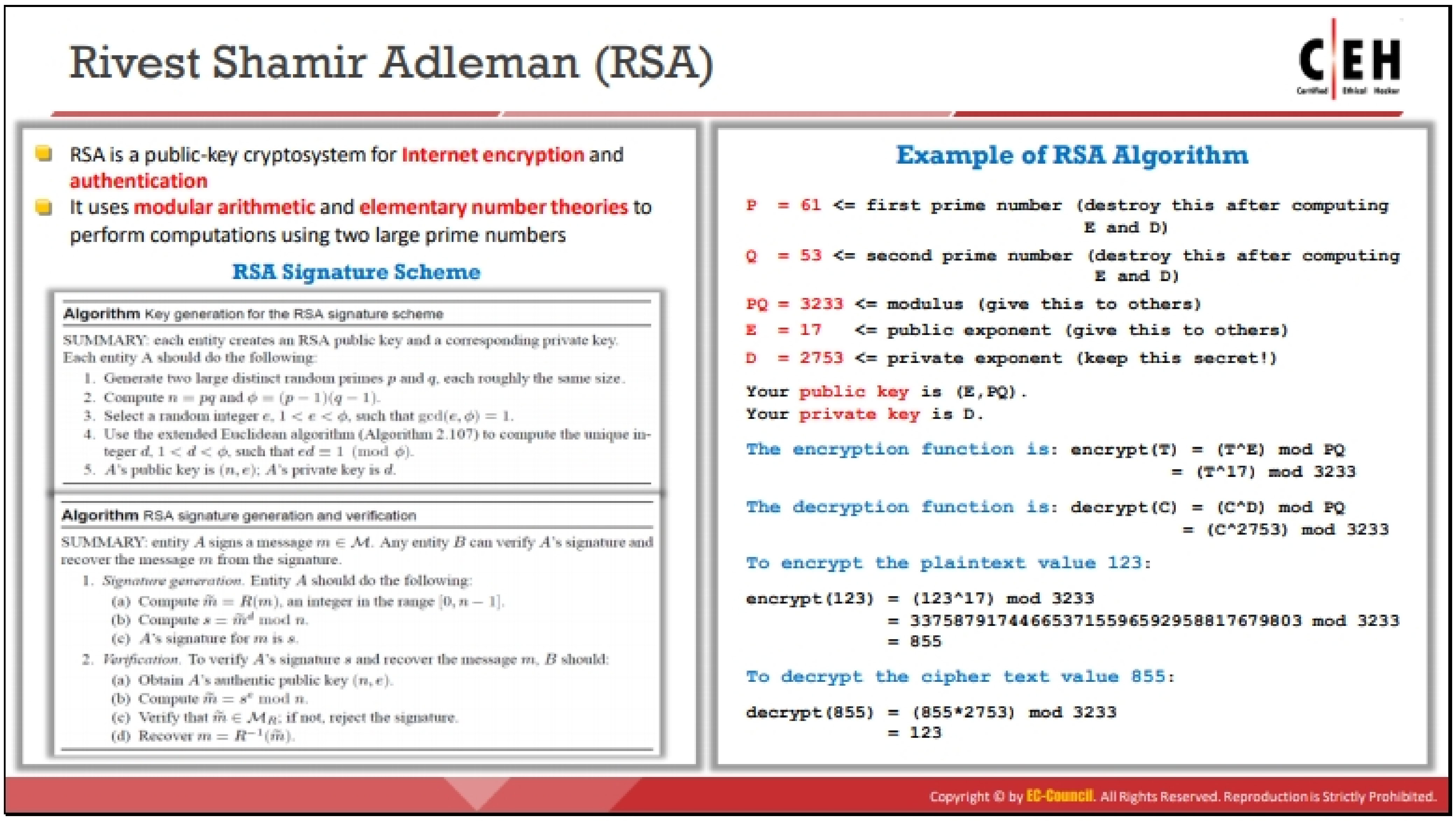
2. Choose t such that $0 \leq t \leq 8$, and select a prime number p where $2^{511+64t} < p < 2^{512+64t}$, with the property that q divides $(p-1)$
3. Select a generator α of the unique cyclic group of order q in Z^*_p by choosing an element $g \in Z^*_p$ and then computing $\alpha = g^{(p-1)/q} \bmod p$ until $\alpha \neq 1$
4. Select a random integer d such that $1 \leq d \leq q-1$
5. Compute $y = \alpha^d \bmod p$
6. A's public key is (p, q, α, y); A's private key is d.

To sign a message m, A does the following:

1. Select a random secret integer k, $0<k<q$.
2. Compute $r = (\alpha^k \bmod p) \bmod q$
3. Compute $k^{-1} \bmod q$
4. Compute $s = k^{-1}\{h(m) + dr\} \bmod q$, where h is the Secure Hash Algorithm
5. A's signature for m is the pair (r, s)

To verify A's signature (r, s) on m, B should do the following:

1. Obtain A's authentic public key (p, q, α, y)
2. Verify that $0<r<q$ and $0<s<q$; if not, then reject the signature
3. Compute $w = s^{-1} \bmod q$ and $h(m)$
4. Compute $u_1 = w.h(m) \bmod q$ and $u_2 = rw \bmod q$
5. Compute $v = (\alpha^{u1} y^{u2} \bmod p) \bmod q$
6. Accept the signature if and only if $v=r$



Rivest Shamir Adleman (RSA)

- RSA is a public-key cryptosystem for **Internet encryption** and **authentication**
- It uses **modular arithmetic** and **elementary number theories** to perform computations using two large prime numbers

RSA Signature Scheme

Algorithm Key generation for the RSA signature scheme

SUMMARY: each entity creates an RSA public key and a corresponding private key. Each entity A should do the following:

1. Generate two large distinct random primes p and q, each roughly the same size.
2. Compute $n = pq$ and $\phi = (p-1)(q-1)$.
3. Select a random integer e, $1 < e < \phi$, such that $\gcd(e, \phi) = 1$.
4. Use the extended Euclidean algorithm (Algorithm 2.107) to compute the unique integer d, $1 < d < \phi$, such that $ed \equiv 1 \pmod{\phi}$.
5. A's public key is (n, e); A's private key is d.

Algorithm RSA signature generation and verification

SUMMARY: entity A signs a message $m \in \mathcal{M}$. Any entity B can verify A's signature and recover the message m from the signature.

1. *Signature generation.* Entity A should do the following:
    (a) Compute $\tilde{m} = R(m)$, an integer in the range $[0, n-1]$.
    (b) Compute $s = \tilde{m}^d \bmod n$.
    (c) A's signature for m is s.
2. *Verification.* To verify A's signature s and recover the message m, B should:
    (a) Obtain A's authentic public key (n, e).
    (b) Compute $\tilde{m} = s^e \bmod n$.
    (c) Verify that $\tilde{m} \in \mathcal{M}_R$; if not, reject the signature.
    (d) Recover $m = R^{-1}(\tilde{m})$.

Example of RSA Algorithm

```
P  = 61 <= first prime number (destroy this after computing
                                 E and D)
Q  = 53 <= second prime number (destroy this after computing
                                 E and D)
PQ = 3233 <= modulus (give this to others)
E  = 17   <= public exponent (give this to others)
D  = 2753 <= private exponent (keep this secret!)

Your public key is (E,PQ).
Your private key is D.

The encryption function is: encrypt(T) = (T^E) mod PQ
                                       = (T^17) mod 3233

The decryption function is: decrypt(C) = (C^D) mod PQ
                                        = (C^2753) mod 3233

To encrypt the plaintext value 123:

encrypt(123) = (123^17) mod 3233
             = 337587917446653715596592958817679803 mod 3233
             = 855

To decrypt the cipher text value 855:

decrypt(855) = (855*2753) mod 3233
             = 123
```

Rivest Shamir Adleman (RSA)

Ron Rivest, Adi Shamir, and Leonard Adleman formulated RSA, a public-key cryptosystem for Internet encryption and authentication. RSA uses modular arithmetic and elementary number theories to perform computations using two large prime numbers. The RSA system is widely used in a variety of products, platforms, and industries. It is one of the de-facto encryption standards. Companies such as Microsoft, Apple, Sun, and Novell build RSA algorithms into their operating systems. RSA can also be found on hardware-secured telephones, Ethernet network cards, and smart cards.

RSA works as follows:

1. Two large prime numbers are taken (a and b), and their product is determined (c = ab, where “c” is called the modulus).
2. RSA chooses a number “e” that it is less than “c” and relatively prime to (a-1)(b-1). Therefore, e and (a-1)(b-1) have no common factor except 1.
3. Furthermore, RSA chooses a number “f” such that (ef - 1) is divisible by (a-1)(b-1).
4. The values “e” and “f” are the public and private exponents, respectively.
5. The public key is the pair (c, e); the private key is the pair (c, f).
6. It is difficult to obtain the private key (c, f) from the public key (c, e). However, if someone can factor “c” into “a” and “b”, then that person can decipher the private key (c, f).

The security of the RSA system depends on the assumption that such factoring is difficult to carry out, making the cryptographic technique safe.

An example of how cryptography uses RSA algorithms in a practical interchange is illustrated by the following sequence:

1. The sender of a message encrypts it using a randomly chosen DES symmetric key. DES (Data Encryption Standard) is a relatively insecure symmetric-key system using 64-bit encryption (56 bits for key size, 8 bits for cyclic redundancy check) to encrypt data.
2. The sender will then look up the recipient's public key and use it to encrypt the DES key using the RSA system.
3. The sender transmits an RSA digital envelope, consisting of a DES-encrypted message and an RSA-encrypted DES key, to the recipient.
4. The recipient will decrypt the DES key and then use the DES key to decrypt the message itself.

This system combines the high speed of DES with the key management convenience of the RSA system.

RSA Signature Scheme

Cryptography uses RSA for public key encryption and for a digital signature (to sign a message and verify it). The RSA signature scheme is the first technique used to generate digital signatures. It is a deterministic digital signature scheme that provides message recovery from the signature itself, making it the most practical and versatile technique available.

RSA involves both a public key and a private key. The public key, as the name indicates, can be used by anyone for encrypting messages. The messages that the user encrypts with the public key require the private key for decryption.

Consider that John encrypts his document M using his private key S_A, thereby creating a signature $S_{john}(M)$. John sends M along with the signature $S_{john}(M)$ to Alice. Alice decrypts the document using Alice's public key, thereby verifying John's signature.

RSA Key Generation

The procedure for RSA key generation is common to all the RSA-based signature schemes. To generate an RSA key pair, i.e., both an RSA public key and the corresponding private key, each entity A should do the following:

- Generate two large distinct primes p and q arbitrarily, each with roughly the same bit length
- Compute $n = pq$ and $\phi = (p-1)(q-1)$
- Choose a random integer e, $1 < e < \phi$,such that $\gcd(e, \phi) = 1$

  GCD = Greatest Common Divisor
- Use the extended Euclidean algorithm to compute the unique integer d, $1 < d < \phi$,such that $ed \equiv 1 \pmod{\phi}$
- A's public key is (n, e); A's private key is d

Destroy p and q at the end of the key generation

The RSA algorithm generates and verifies the RSA signature as follows way:

Entity A signs a message $m \in M$. Any entity B can verify A's signature and recover the message m from the signature.

1. **Signature Generation**

   To sign a message m, entity A should do the following:

   - Compute $\tilde{m} = R(m)$, an integer in the range [0, n–1]
   - Compute $s = \tilde{m}^d \bmod n$
   - A's signature form is s

2. **Signature Verification**

   To verify A's signature s and recover the message m, B should do the following:

   - Obtain A's authentic public key (n, e)
   - Compute $\tilde{m} = s^e \bmod n$
   - Verify that $\tilde{m} \in M_R$; if not, reject the signature
   - Recover $m = R^{-1}(\tilde{m})$

Example of RSA Algorithm

The math underlying RSA public-key encryption is described below:

1. Find P and Q, two large (e.g., 1024-bit) prime numbers.
2. Choose E such that E is greater than 1, E is less than PQ, and E and (P-1)(Q-1) are relatively prime, which means that they have no prime factors in common. E does not have to be prime, but it must be odd. (P-1)(Q-1) cannot be prime because it is an even number.
3. Compute D such that (DE - 1) is evenly divisible by (P-1)(Q-1). Mathematicians write this as DE = 1 (mod (P-1)(Q-1)), and they call D the multiplicative inverse of E. This is easy to do—simply find an integer X that causes D = (X(P-1)(Q-1) + 1)/E to be an integer and then use that value of D.
4. The encryption function is C = (T^E) mod PQ, where C is the ciphertext (a positive integer), T is the plaintext (a positive integer), and ^ indicates exponentiation. During the encryption of the message, T must be less than the modulus, PQ.
5. The decryption function is T = (C^D) mod PQ, where C is the ciphertext (a positive integer), T is the plaintext (a positive integer), and ^ indicates exponentiation.

Your public key is the pair (PQ, E). Your private key is the number D (do not reveal it to anyone). The product PQ is the modulus. E is the public exponent. D is the secret exponent.

You can publish your public key freely because there are no known easy methods of calculating D, P, or Q given only (PQ, E) (your public key).

Given below is an example of the RSA algorithm:

P = 61<= first prime number (destroy this after computing E and D)

Q = 53<= second prime number (destroy this after computing E and D)

PQ = 3233<= modulus (give this to others)

E = 17<= public exponent (give this to others)

D = 2753<= private exponent (keep this secret)

Your **public** key is **(E,PQ)**

Your **private** key is **D**

The encryption function is:

encrypt(T) = (T^E) mod PQ

= (T^17) mod 3233

The decryption function is:

decrypt(C) = (C^D) mod PQ

= (C^2753) mod 3233

To encrypt the plaintext value 123, do this:

encrypt(123) = (123^17) mod 3233

= 337587917446653715596592958817679803 mod 3233

= 855

To decrypt the ciphertext value 855, do this:

decrypt(855) = (855*2753) mod 3233

= 123

One way to compute the value of **855^2753 mod 3233** is as follows:

Consider these powers of 855:

- 855^1 = 855 (mod 3233)
- 855^2 = 367 (mod 3233)
- 855^4 = 367^2 (mod 3233) = 2136 (mod 3233)
- 855^8 = 2136^2 (mod 3233) = 733 (mod 3233)
- 855^16 = 733^2 (mod 3233) = 611 (mod 3233)
- 855^32 = 611^2 (mod 3233) = 1526 (mod 3233)
- 855^64 = 1526^2 (mod 3233) = 916 (mod 3233)
- 855^128 = 916^2 (mod 3233) = 1709 (mod 3233)

- 855^256 = 1709^2 (mod 3233) = 1282 (mod 3233)
- 855^512 = 1282^2 (mod 3233) = 1160 (mod 3233)
- 855^1024 = 1160^2 (mod 3233) = 672 (mod 3233)
- 855^2048 = 672^2 (mod 3233) = 2197 (mod 3233)

Given the above, we know the following:

855^2753 (mod 3233)

= 855^(1 + 64 + 128 + 512 + 2048) (mod 3233)

= 855^1 * 855^64 * 855^128 * 855^512 * 855^2048 (mod 3233)

= 855 * 916 * 1709 * 1160 * 2197 (mod 3233)

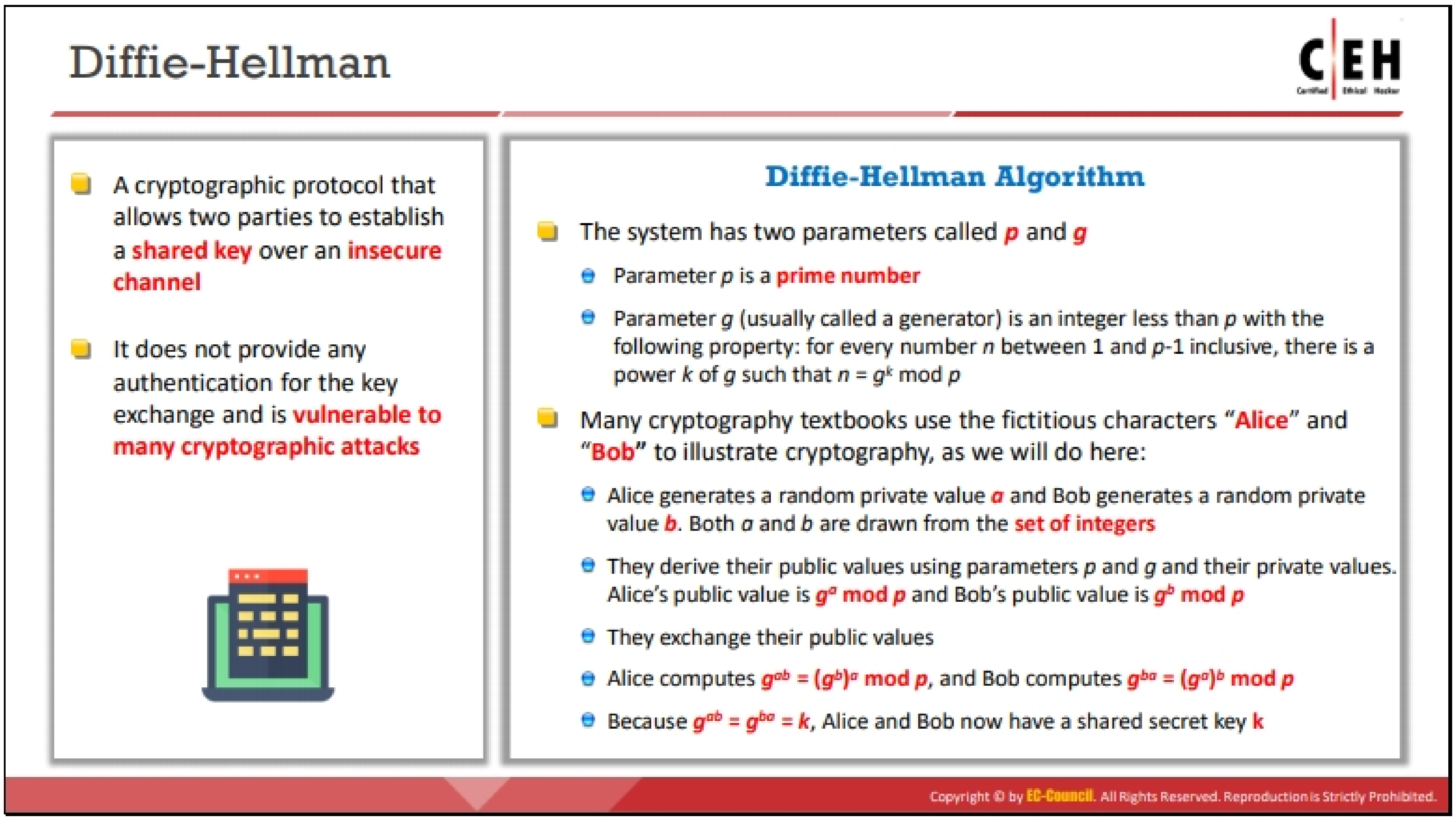
= 794 * 1709 * 1160 * 2197 (mod 3233)

= 2319 * 1160 * 2197 (mod 3233)

= 184 * 2197 (mod 3233)

= 123 (mod 3233)

= 123



Diffie-Hellman

- A cryptographic protocol that allows two parties to establish a **shared key** over an **insecure channel**
- It does not provide any authentication for the key exchange and is **vulnerable to many cryptographic attacks**

Diffie-Hellman Algorithm

- The system has two parameters called p and g
  - Parameter p is a **prime number**
  - Parameter g (usually called a generator) is an integer less than p with the following property: for every number n between 1 and p-1 inclusive, there is a power k of g such that $n = g^k \bmod p$
- Many cryptography textbooks use the fictitious characters "**Alice**" and "**Bob**" to illustrate cryptography, as we will do here:
  - Alice generates a random private value a and Bob generates a random private value b. Both a and b are drawn from the **set of integers**
  - They derive their public values using parameters p and g and their private values. Alice's public value is $g^a \bmod p$ and Bob's public value is $g^b \bmod p$
  - They exchange their public values
  - Alice computes $g^{ab} = (g^b)^a \bmod p$, and Bob computes $g^{ba} = (g^a)^b \bmod p$
  - Because $g^{ab} = g^{ba} = k$, Alice and Bob now have a shared secret key **k**

Diffie–Hellman

It is a cryptographic protocol that allows two parties to establish a shared key over an insecure channel. It was developed and published by Whitfield Diffie and Martin Hellman in 1976. Actually, it was independently developed a few years earlier by Malcolm J. Williamson of the British Intelligence Service, but it was classified at that time.

Diffie–Hellman Algorithm

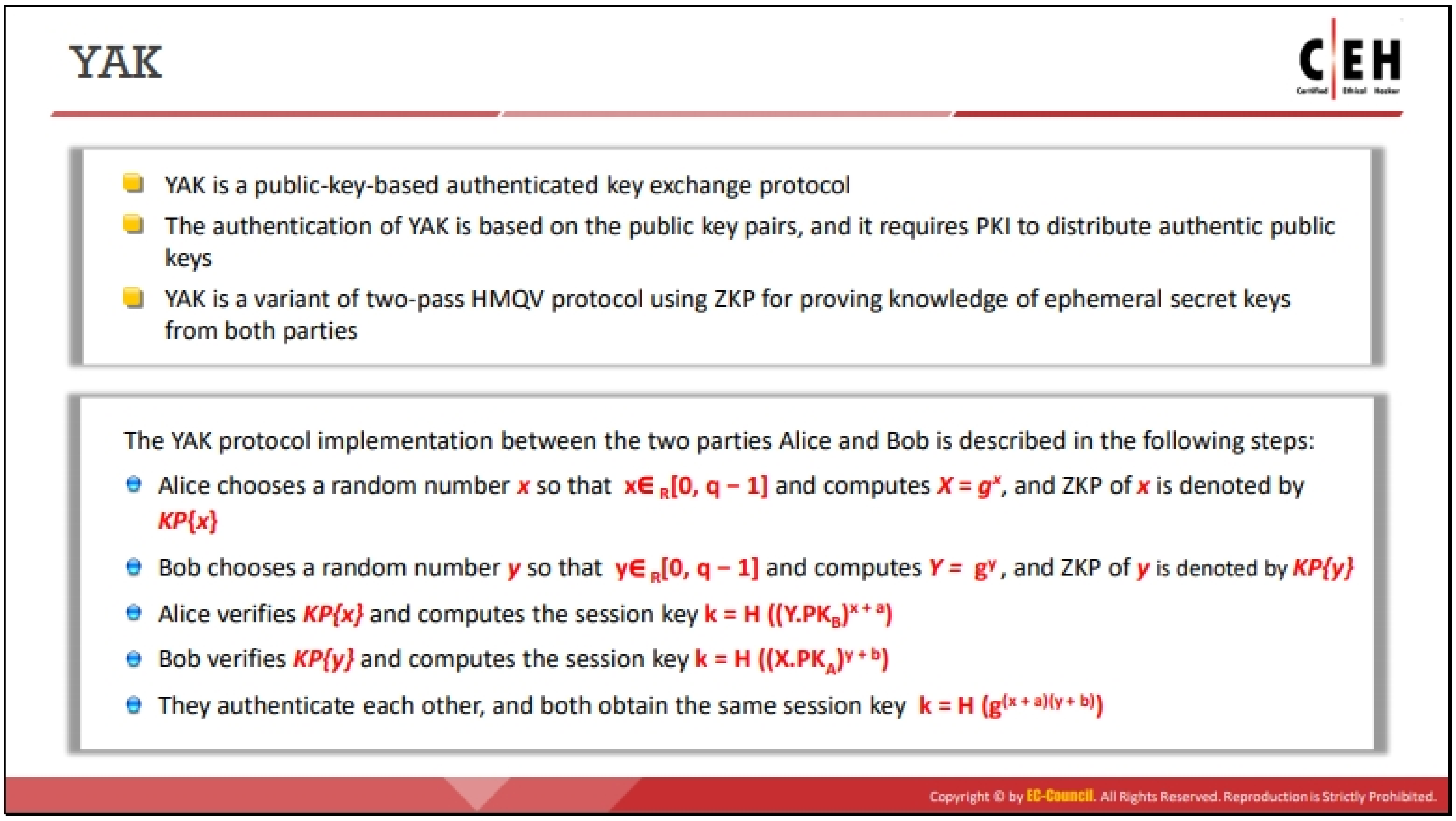
The system has two parameters called p and g

- Parameter p is a prime number
- Parameter g (usually called a generator) is an integer less than p, with the following property: for every number n between 1 and $p-1$ (both inclusive), there is a power k of g such that $n = g^k \bmod p$

Many cryptography textbooks use the fictitious characters "Alice" and "Bob" to illustrate cryptography; we will do the same here as well:

- Alice generates a random private value **a**, and Bob generates a random private value **b**. Both a and b are drawn from the **set of integers**
- They derive their public values using parameters p and g and their private values. Alice's public value is $g^a \bmod p$, and Bob's public value is $g^b \bmod p$.
- They exchange their public values
- Alice computes $g^{ab} = (g^b)^a \bmod p$, and Bob computes $g^{ba} = (g^a)^b \bmod p$
- Since $g^{ab} = g^{ba} = k$, Alice and Bob now have a shared secret key **k**

The Diffie–Hellman algorithm does not provide any authentication for the key exchange and is vulnerable to many cryptographic attacks. Nevertheless, it is the basis of many authentication mechanisms; for example, it provides forward secrecy in the TLS protocol's ephemeral modes depending on the cipher spec.



YAK

- YAK is a public-key-based authenticated key exchange protocol
- The authentication of YAK is based on the public key pairs, and it requires PKI to distribute authentic public keys
- YAK is a variant of two-pass HMQV protocol using ZKP for proving knowledge of ephemeral secret keys from both parties

The YAK protocol implementation between the two parties Alice and Bob is described in the following steps:

- Alice chooses a random number x so that $x \in_R [0, q-1]$ and computes $X = g^x$, and ZKP of x is denoted by $KP\{x\}$
- Bob chooses a random number y so that $y \in_R [0, q-1]$ and computes $Y = g^y$, and ZKP of y is denoted by $KP\{y\}$
- Alice verifies $KP\{x\}$ and computes the session key $k = H((Y.PK_B)^{x+a})$
- Bob verifies $KP\{y\}$ and computes the session key $k = H((X.PK_A)^{y+b})$
- They authenticate each other, and both obtain the same session key $k = H(g^{(x+a)(y+b)})$

YAK

YAK is a public-key-based Authenticated Key Exchange (AKE) protocol. The authentication of YAK is based on public key pairs, and it needs PKI to distribute authentic public keys. YAK is a variant of the two-pass Hashed Menezes-Qu-Vanstone (HMQV) protocol using zero-knowledge proofs (ZKP) for proving the knowledge of ephemeral secret keys from both parties. The YAK protocol lacks joint key control and perfect forward secrecy attributes.

The YAK protocol implementation between two parties Alice and Bob is described as follows:

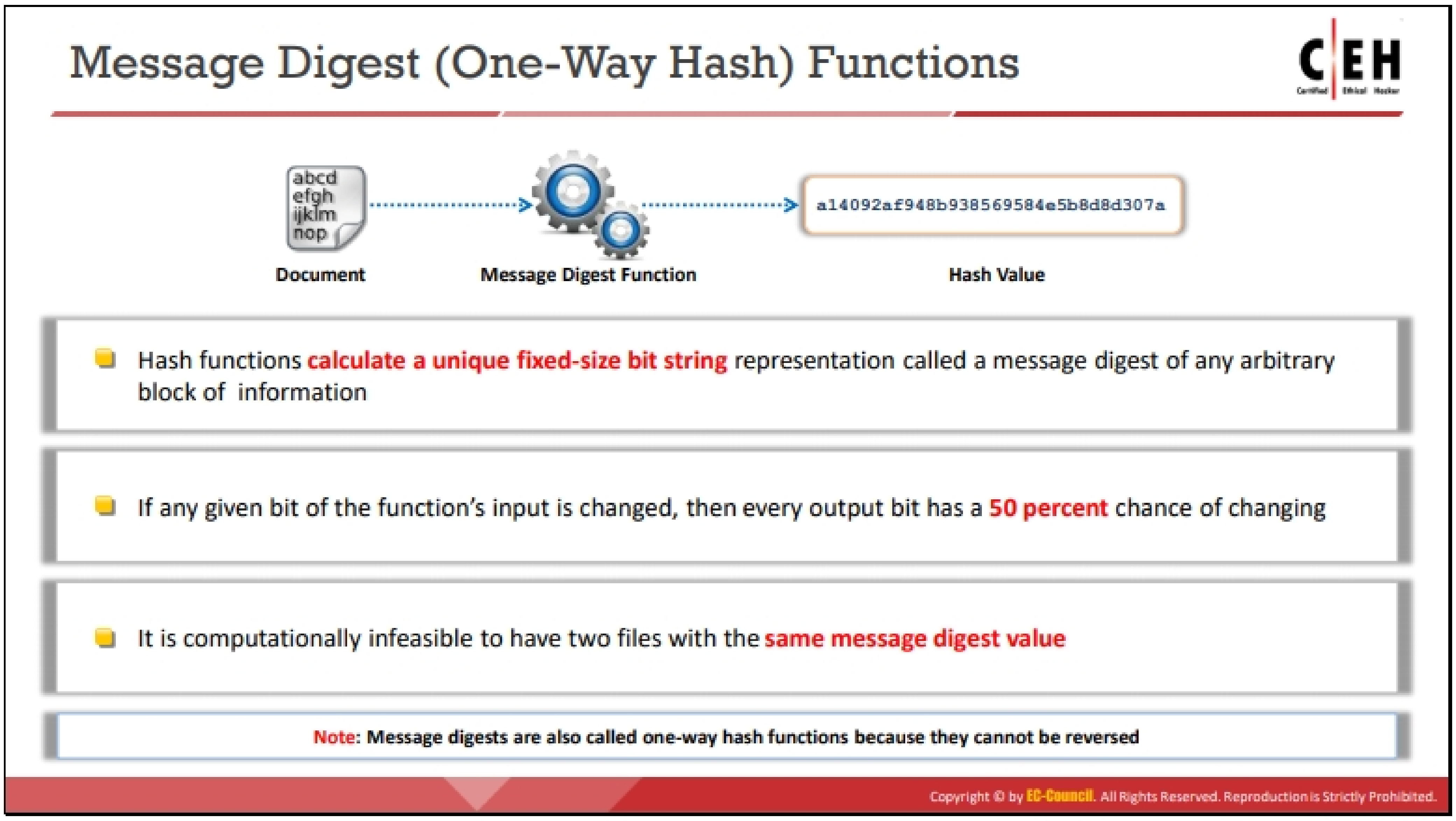
1. Alice chooses a random number **x** such that $x \in_R [0, q-1]$, computes $X = g^x$, and generates ZKP of **x**, denoted by **KP{x}**. Alice sends **X** and **KP{x}** to Bob.
2. Bob chooses a random number **y** such that $y \in_R [0, q-1]$, computes $Y = g^y$, and generates ZKP of **y**, denoted by **KP{y}**. Bob sends **Y** and **KP{y}** to Alice.
3. Alice verifies the received **KP{x}** and computes the session key after verification as

   $k = H((Y.PK_B)^{x+a})$, where **H** is a hash function.
4. Bob verifies the received **KP{y}** and computes the session key after verification as

   $k = H((X.PK_A)^{y+b})$.
5. They authenticate each other, and both obtain the same session key $k = H(g^{(x+a)(y+b)})$.

The YAK protocol can accomplish the following objectives:

- Private key security
- Full forward secrecy
- Session key security



Message Digest (One-Way Hash) Functions

- Hash functions **calculate a unique fixed-size bit string** representation called a message digest of any arbitrary block of information
- If any given bit of the function's input is changed, then every output bit has a **50 percent** chance of changing
- It is computationally infeasible to have two files with the **same message digest value**

Note: Message digests are also called one-way hash functions because they cannot be reversed

Message Digest (One-way Hash) Functions

Hash functions calculate a unique fixed-size bit string representation, called a message digest, of any arbitrary block of information. Message digest functions distill the information contained in a file (small or large) into a single fixed-length number, typically between 128 and 256 bits. If any given bit of the function's input is changed, every output bit has a 50% chance of changing. Given an input file and its corresponding message digest, it should be nearly impossible to find another file with the same message digest value, as it is computationally infeasible to have two files with the same message digest value.

Message digest functions are also called one-way hash functions because they produce values that are nearly impossible to invert, resistant to attack, mostly unique, and widely distributed. Message digest algorithms themselves do not participate in encryption and decryption operations. They allow the creation of digital signatures and message authentication codes (MACs) as well as the derivation of encryption keys from passphrases.

The main role of a cryptographic hash function is to provide integrity in document management. Cryptographic hash functions are an integral part of digital signatures. They are relatively faster than digital signature algorithms; hence, their characteristic feature is to calculate the signature of the document's hash value, which is smaller than the document. In addition, digests help to hide the contents or source of the document.

Widely used message digest functions include the following algorithms:

- MD5
- SHA

Note: Message digests are also called one-way hash functions because they cannot be reversed.

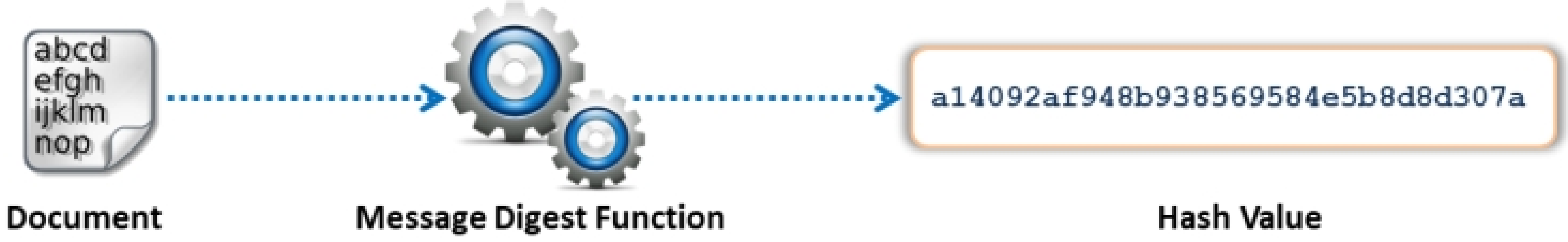
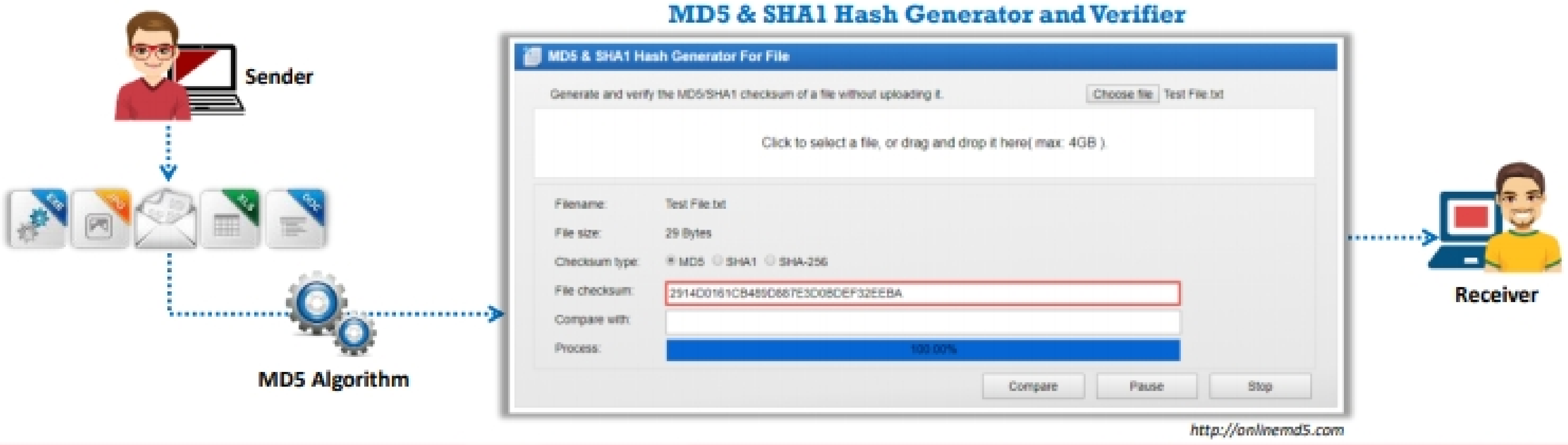

abcd
efgh
ijklm
nop
a14092af948b938569584e5b8d8d307a
Document
Message Digest Function
Hash Value

Figure 20.8: Block Diagram of One-way Hash Function

Message Digest Function: MD5 and MD6

- The MD5 algorithm takes a message of **arbitrary length** as the input and then outputs a **128-bit fingerprint** or message digest of the input
- MD5 is not collision resistant; use of the latest algorithms, such as **MD6**, **SHA-2** and **SHA-3**, is recommended
- **MD6** uses a Merkle tree-like structure to allow for immense parallel computation of hashes for very long inputs. It is resistant to differential cryptanalysis attacks
- MD5 and MD6 are deployed for digital signature applications, file integrity checking, and storing passwords

MD5 & SHA1 Hash Generator and Verifier



Generate and verify the MD5/SHA1 checksum of a file without uploading it. Test File.txt

Click to select a file, or drag and drop it here(max: 4GB).

Filename: Test File.txt
File size: 29 Bytes
Checksum type: ● MD5 ○ SHA1 ○ SHA-256
File checksum: 2914D0161CB489D887E3D08DEF32EEBA
Compare with:
Process: 100.00%

http://onlinemd5.com

Message Digest Function: MD5 and MD6

MD2, MD4, MD5, and MD6 are **message digest algorithms** used in digital signature applications to compress a document securely before the system signs it with a private key. The algorithms can be of variable length, but the resulting message digest always has a size of 128 bits.

The structures of all three algorithms (MD2, MD4, and MD5) appear similar, although the design of MD2 is reasonably different from that of MD4 and MD5. MD2 supports 8-bit machines, while MD4 and MD5 support 32-bit machines. The algorithm pads the message with extra bits to ensure that the number of bits is divisible by 512. The extra bits may include a 64-bit binary message.

Attacks on versions of MD4 have become increasingly successful. Research has shown how an attacker launches collision attacks on the full version of MD4 within a minute on a typical PC. MD5 is slightly more secure but is slower than MD4. However, both the message digest size and the padding requirements remain the same.

MD5 is a widely used cryptographic hash function that takes a message of arbitrary length as input and outputs a 128-bit (16-byte) fingerprint or message digest of the input. MD5 can be used in a wide variety of cryptographic applications and is useful for digital signature applications, file integrity checking, and storing passwords. However, MD5 is not collision resistant; therefore, it is better to use the latest algorithms, such as MD6, SHA-2, and SHA-3.

MD6 uses a Merkle-tree-like structure to allow for large-scale parallel computation of hashes for very long inputs. It is resistant to differential cryptanalysis attacks.

To calculate the effectiveness of hash functions, check the output produced when the algorithm randomizes an arbitrary input message.

The following are examples of minimally different message digests:

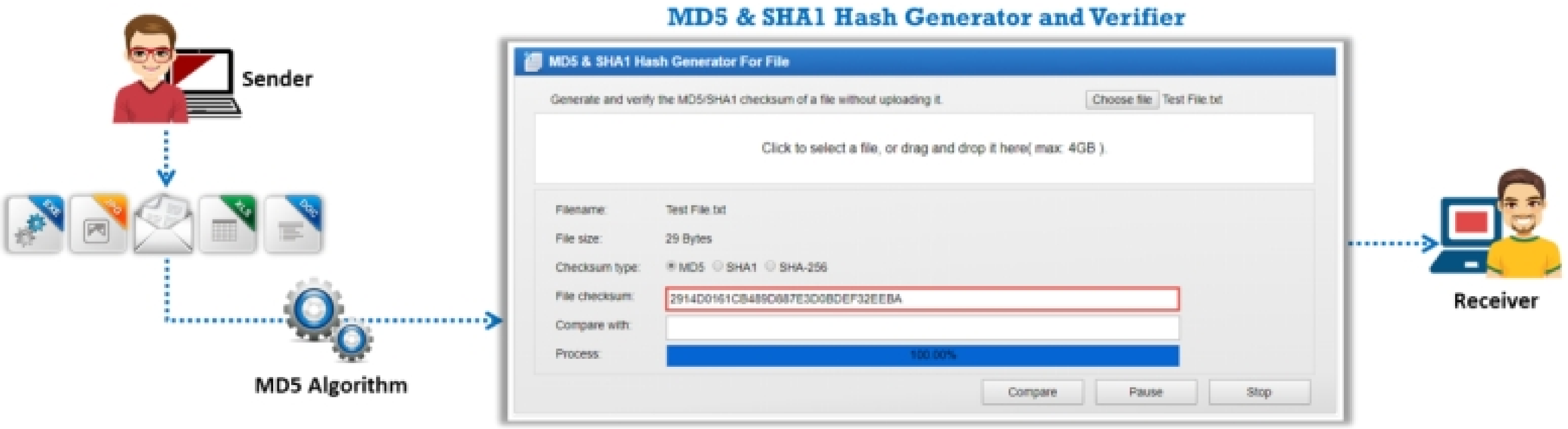
- echo “There is CHF1500 in the blue bo” | md5sum

  e41a323bdf20eadafd3f0e4f72055d36
- echo “There is CHF1500 in the blue box” | md5sum

  7a0da864a41fd0200ae0ae97afd3279d
- echo “There is CHF1500 in the blue box.” | md5sum

  2db1ff7a70245309e9f2165c6c34999d

Even minimally different texts produce radically different MD5 codes.


MD5 & SHA1 Hash Generator and Verifier
Sender
MD5 Algorithm
Receiver
MD5 & SHA1 Hash Generator For File
Choose file
Compare
Pause
Stop

Generate and verify the MD5/SHA1 checksum of a file without uploading it. Test File.txt

Click to select a file, or drag and drop it here(max: 4GB).

Filename: Test File.txt

File size: 29 Bytes

Checksum type: ● MD5 ○ SHA1 ○ SHA-256

File checksum: 2914D0161CB489D887E3D0BDEF32EEBA

Compare with:

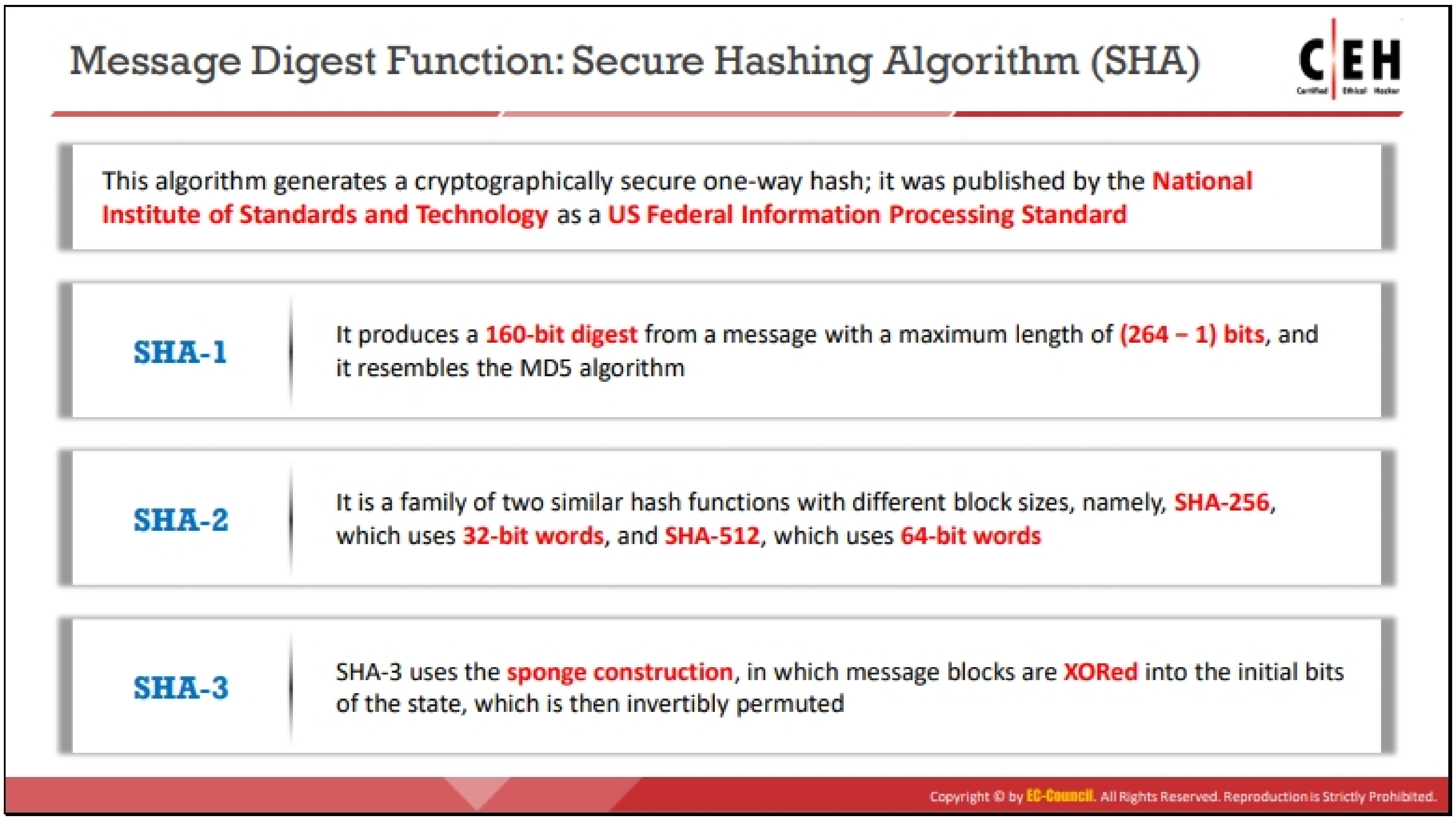
Process: 100.00%

Figure 20.9: Verifying MD5 Hash

- **Onlinemd5**

  Source: *http://onlinemd5.com*

  Onlinemd5 generates and checks file integrity using secure time-proven algorithms such as MD5, SHA-1, and SHA-256. One can create checksums (digital fingerprints) of files and verify their integrity using this online tool.



Message Digest Function: Secure Hashing Algorithm (SHA)

This algorithm generates a cryptographically secure one-way hash; it was published by the **National Institute of Standards and Technology** as a **US Federal Information Processing Standard**

SHA-1	It produces a **160-bit digest** from a message with a maximum length of **(264 – 1) bits**, and it resembles the MD5 algorithm
SHA-2	It is a family of two similar hash functions with different block sizes, namely, **SHA-256**, which uses **32-bit words**, and **SHA-512**, which uses **64-bit words**
SHA-3	SHA-3 uses the **sponge construction**, in which message blocks are **XORed** into the initial bits of the state, which is then invertibly permuted

Message Digest Function: Secure Hashing Algorithm (SHA)

The NIST has developed the Secure Hash Algorithm (SHA), specified in the **Secure Hash Standard** (**SHS**) and published as a federal information-processing standard (FIPS PUB 180). It generates a cryptographically secure one-way hash. Rivest developed the SHA, which is similar to the message digest algorithm family of hash functions. It is slightly slower than MD5, but its larger message digest makes it more secure against brute-force collision and inversion attacks.

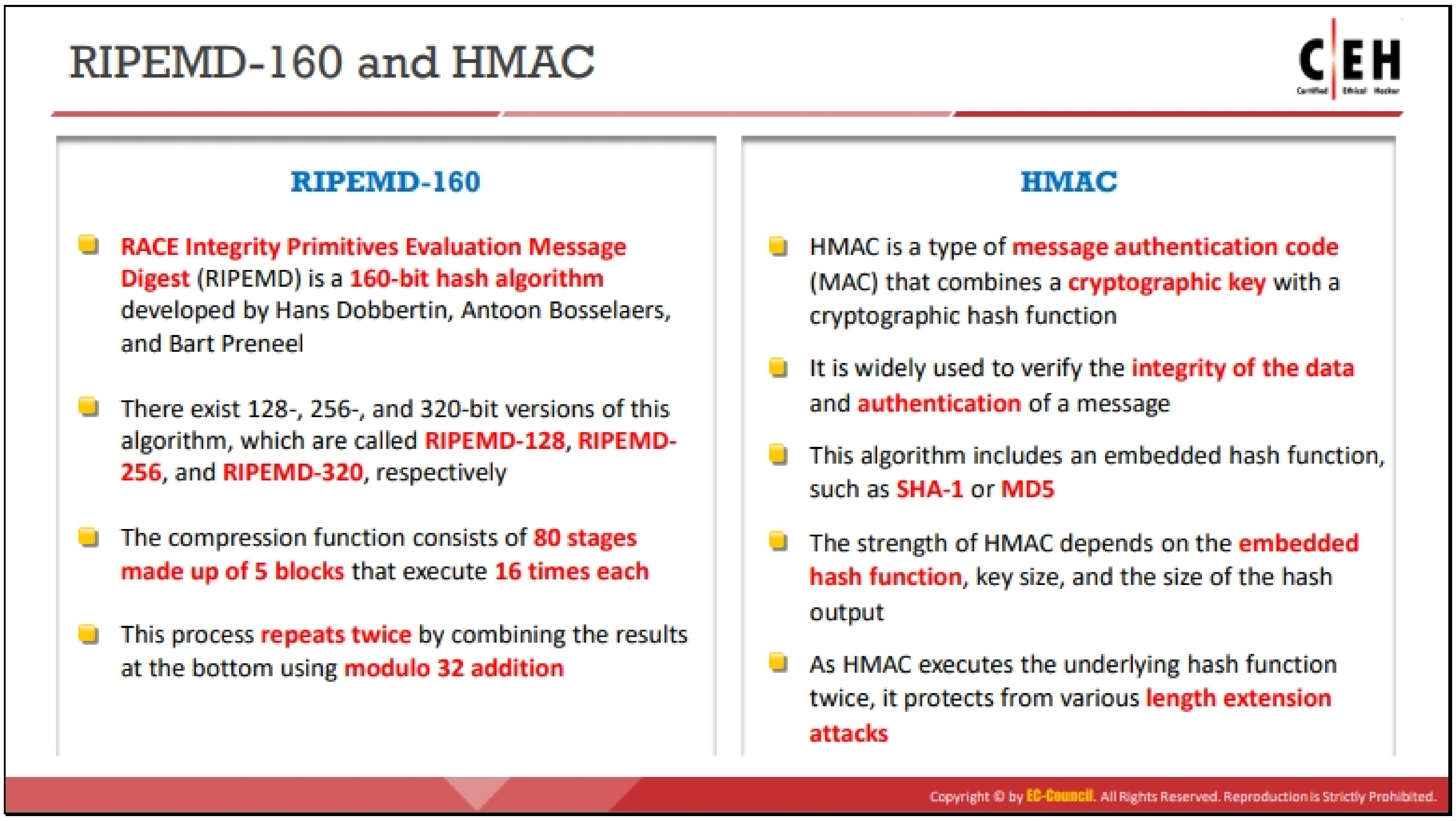
SHA encryption is a series of five different cryptographic functions, and it currently has three generations: SHA-1, SHA-2, and SHA-3.

- **SHA-0**: A retronym applied to the original version of the 160-bit hash function published in 1993 under the name SHA, which was withdrawn from trade due to an undisclosed "**significant flaw**" in it. It was replaced with a slightly revised version, namely SHA-1.
- **SHA-1**: It is a 160-bit hash function that resembles the former MD5 algorithm developed by Ron Rivest. It produces a 160-bit digest from a message with a maximum length of (264 − 1) bits. It was designed by the National Security Agency (NSA) to be part of the Digital Signature Algorithm (DSA). It is most commonly used in security protocols such as PGP, TLS, SSH, and SSL. As of 2010, SHA-1 is no longer approved for cryptographic use because of its cryptographic weaknesses.
- **SHA-2**: SHA2 is a family of two similar hash functions with different block sizes, namely SHA-256, which uses 32-bit words, and SHA-512, which uses 64-bit words. The truncated versions of each standard are SHA-224 and SHA-384.
- **SHA-3**: SHA-3 uses sponge construction in which message blocks are XORed into the initial bits of the state, which the algorithm then invertibly permutes. It supports the same hash lengths as SHA-2 but differs in its internal structure considerably from the rest of the SHA family.

Comparison of SHA functions (SHA-0, SHA-1, SHA-2, and SHA-3).

Algorithm and variant		**Output size (bits)**	**Internal state size (bits)**	**Block Size (bits)**	**Maximum message size (bits)**	**Rounds**	**Operations**	**Security (bits)**
MD5 (as reference)		128	128 (4*32)	512	2^{64}-1	64	Add mod 2^{32}, and, or, xor, rot	<=18 (collisions found)
SHA-0		160	160 (5*32)	512	2^{64}-1	80	Add mod 2^{32}, and, or, xor, rot	<34 (collisions found)
SHA-1		160	160 (5*32)	512	2^{64}-1	80	Add mod 2^{32}, and, or, xor, rot	<63 (collisions found)
SHA-2	SHA-224	224	256 (8*32)	512	2^{64}-1	64	Add mod 2^{32}, and, or, xor, shr, rot	112
	SHA-256	256						128
	SHA-384	384	512 (8*64)	1024	2^{128}-1	80	Add mod 2^{64}, and, or, xor, shr, rot.	192
	SHA-512	512						256
	SHA-512/224	224						112
	SHA-512/256	256						128
SHA-3	SHA3-224	224	1600 (5*5*64)	1152	∞	24	And, xor, not, rot	112
	SHA3-256	256		1088				128
	SHA3-384	384		832				192
	SHA3-512	512		576				256
	SHAKE128	d(arbitrary)		1344				Min(d/2,128)
	sHAKE256	d(arbitrary)		1088				Min(d/2,256)

Table 20.2: Comparison between SHA-0, SHA-1 and SHA-2 functions



RIPEMD-160 and HMAC

RIPEMD-160

- **RACE Integrity Primitives Evaluation Message Digest** (RIPEMD) is a **160-bit hash algorithm** developed by Hans Dobbertin, Antoon Bosselaers, and Bart Preneel
- There exist 128-, 256-, and 320-bit versions of this algorithm, which are called **RIPEMD-128**, **RIPEMD-256**, and **RIPEMD-320**, respectively
- The compression function consists of **80 stages made up of 5 blocks** that execute **16 times each**
- This process **repeats twice** by combining the results at the bottom using **modulo 32 addition**

HMAC

- HMAC is a type of **message authentication code** (MAC) that combines a **cryptographic key** with a cryptographic hash function
- It is widely used to verify the **integrity of the data** and **authentication** of a message
- This algorithm includes an embedded hash function, such as **SHA-1** or **MD5**
- The strength of HMAC depends on the **embedded hash function**, key size, and the size of the hash output
- As HMAC executes the underlying hash function twice, it protects from various **length extension attacks**

RIPEMD-160

RACE Integrity Primitives Evaluation Message Digest (RIPEMD) is a 160-bit hash algorithm developed by Hans Dobbertin, Antoon Bosselaers, and Bart Preneel. There exist 128-, 256-, and 320-bit versions of this algorithm, called RIPEMD-128, RIPEMD-256, and RIPEMD-320, respectively. These algorithms replace the original RIPEMD, which was found to have a collision issue. They do not follow any standard security policies or guidelines.

RIPEMD-160 is a more secure version of the RIPEMED algorithm. In this algorithm, the compression function consists of 80 stages, i.e., 5 blocks that execute 16 times each. This process repeats twice by combining the results at the bottom using modulo 32 addition.

HMAC

Hash-based message authentication code (HMAC) is a type of message authentication code (MAC) that uses a cryptographic key along with a cryptographic hash function. It is widely used to verify the integrity of data and authentication of a message. This algorithm includes an embedded hash function such as SHA-1 or MD5. The strength of HMAC depends on the embedded hash function, key size, and size of the hash output.

HMAC includes two stages for computing the hash. The input key is processed to produce two keys, namely the inner key and the outer key. The first stage of the algorithm inputs the inner key and message to produce an internal hash. The second stage of the algorithm inputs the output from the first stage and outer key, and produces the final HMAC code.

As HMAC executes the underlying hash function twice, it offers protection against various length extension attacks. The size of the key and the output depends on the embedded hash function; e.g., 128 or 160 bits in the case of MD5 or SHA-1, respectively.

CHAP

The Challenge-Handshake Authentication Protocol (CHAP) is an authentication mechanism used by Point-to-Point Protocol (PPP) servers to authenticate or validate the identity of remote clients or network hosts. It is more secure and effective compared to Password Authentication Procedure (PAP), as it regularly verifies the identity of the client using a three-way handshake and provides protection against replay attacks.

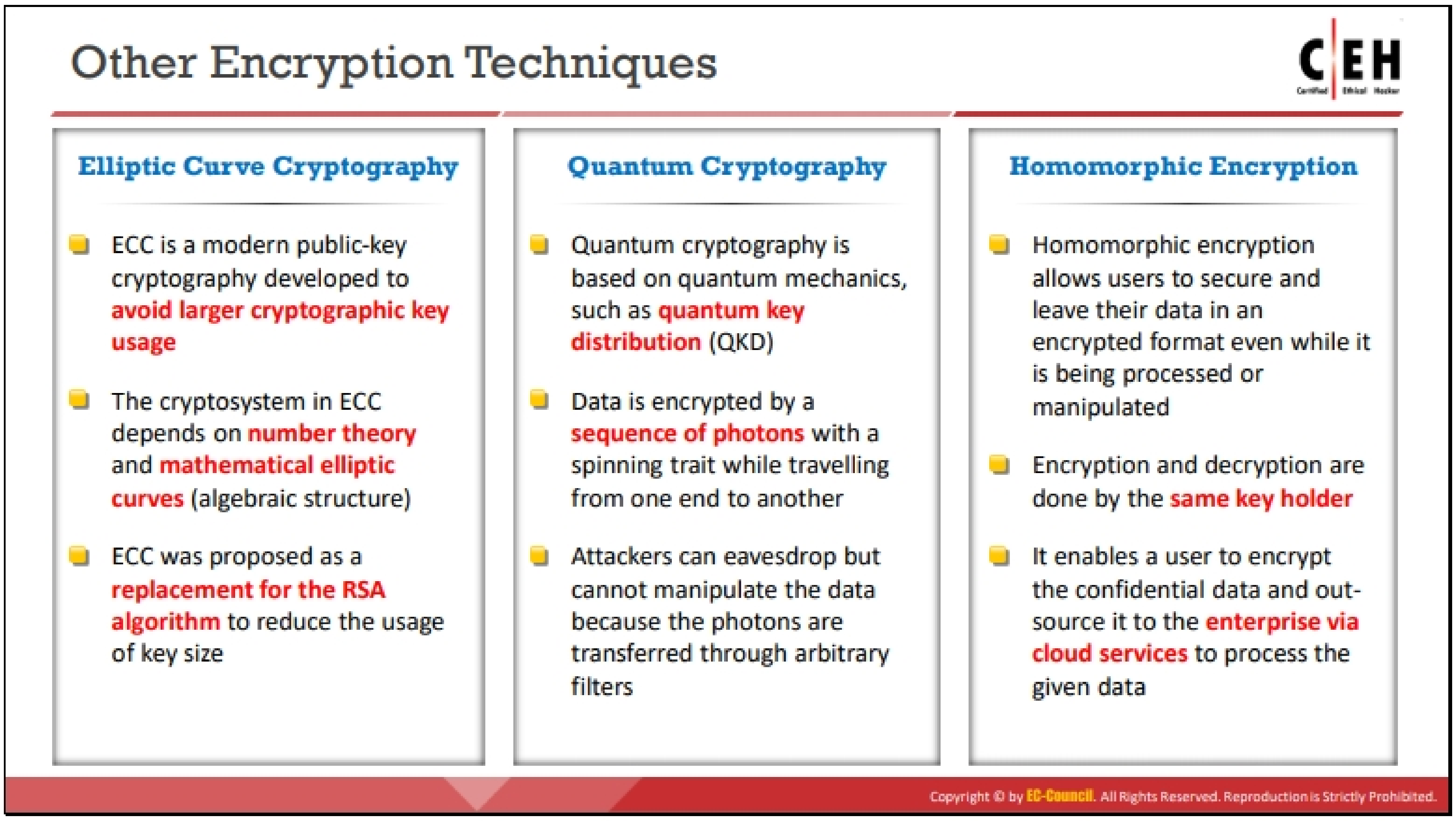
EAP

The Extensible Authentication Protocol (EAP) is an authentication protocol that was originally designed for point-to-point connections. It is used as an alternative to the CHAP and PAP authentication protocols, as it is more secure and supports different authentication mechanisms such as passwords, smart tokens, one-time passwords (OTPs), secure ID card, digital certificates, and public-key encryption mechanisms. After the selection of the EAP authentication mechanism, a session is established and messages are exchanged between the client and the authenticating server. The session consists of requests and responses for authentication information. The length and details of the authentication session are determined by the EAP authentication mechanism used.

GOST – Hash Function

This hash algorithm was initially defined in the Russian national standard GOST R 34.11-94 "Information Technology - Cryptographic Information Security - Hash Function."

It produces a fixed-length output of 256 bits. The input message is broken up into chunks of 256-bit blocks. If a block is less than 256 bits, then the message is padded by appending as many zeros to it as are required to make the length of the message 256 bits. The remaining bits are filled with a 256-bit integer arithmetic sum of all previously hashed blocks. Then, a 256-bit integer representing the length of the original message, in bits, is produced.



Other Encryption Techniques

Elliptic Curve Cryptography

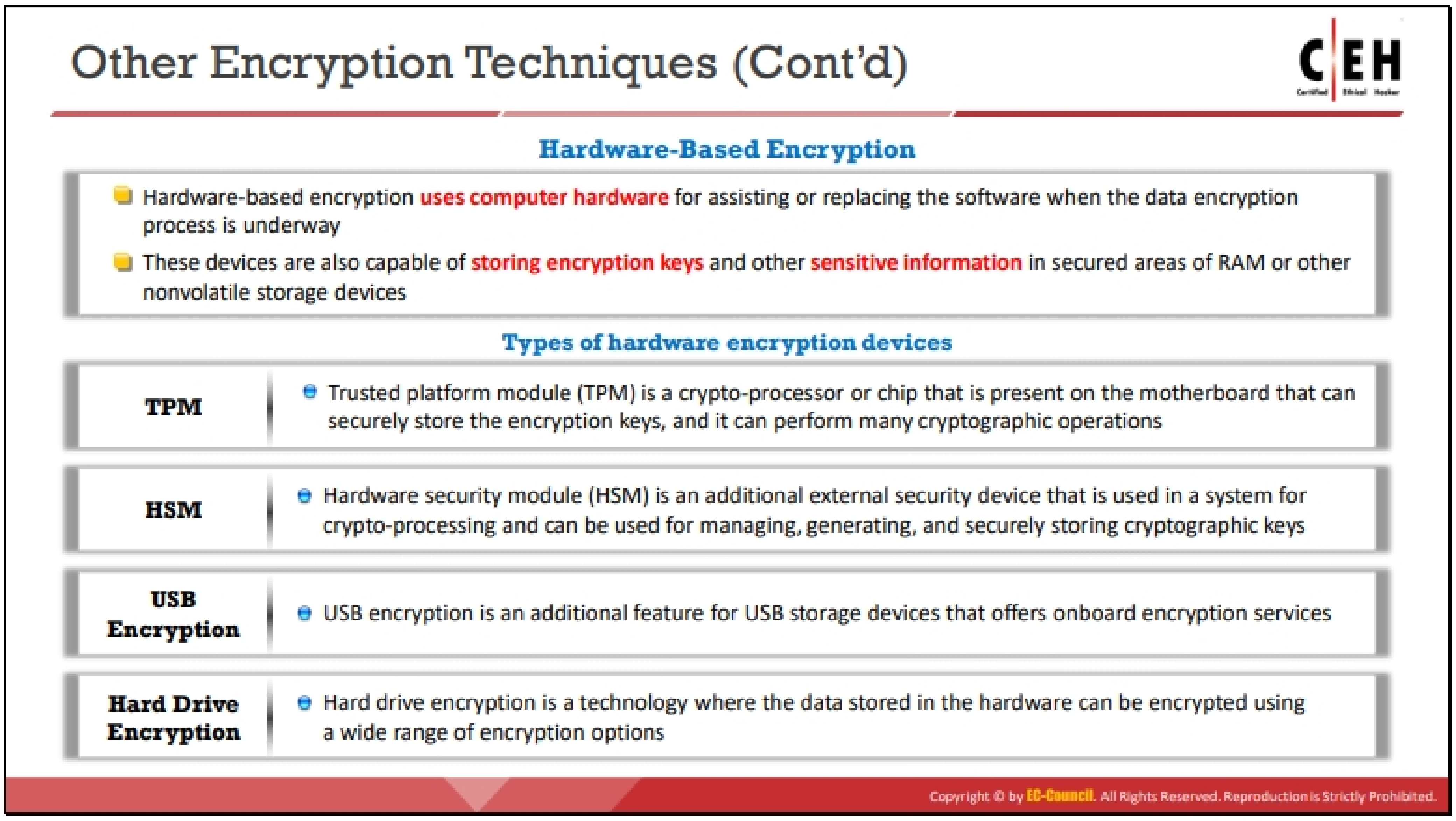
- ECC is a modern public-key cryptography developed to **avoid larger cryptographic key usage**
- The cryptosystem in ECC depends on **number theory** and **mathematical elliptic curves** (algebraic structure)
- ECC was proposed as a **replacement for the RSA algorithm** to reduce the usage of key size

Quantum Cryptography

- Quantum cryptography is based on quantum mechanics, such as **quantum key distribution** (QKD)
- Data is encrypted by a **sequence of photons** with a spinning trait while travelling from one end to another
- Attackers can eavesdrop but cannot manipulate the data because the photons are transferred through arbitrary filters

Homomorphic Encryption

- Homomorphic encryption allows users to secure and leave their data in an encrypted format even while it is being processed or manipulated
- Encryption and decryption are done by the **same key holder**
- It enables a user to encrypt the confidential data and out-source it to the **enterprise via cloud services** to process the given data



Other Encryption Techniques (Cont'd)

Hardware-Based Encryption

- Hardware-based encryption **uses computer hardware** for assisting or replacing the software when the data encryption process is underway
- These devices are also capable of **storing encryption keys** and other **sensitive information** in secured areas of RAM or other nonvolatile storage devices

Types of hardware encryption devices

TPM	Trusted platform module (TPM) is a crypto-processor or chip that is present on the motherboard that can securely store the encryption keys, and it can perform many cryptographic operations
HSM	Hardware security module (HSM) is an additional external security device that is used in a system for crypto-processing and can be used for managing, generating, and securely storing cryptographic keys
USB Encryption	USB encryption is an additional feature for USB storage devices that offers onboard encryption services
Hard Drive Encryption	Hard drive encryption is a technology where the data stored in the hardware can be encrypted using a wide range of encryption options

Other Encryption Techniques

- **Elliptic Curve Cryptography (ECC)**

  ECC is a modern public-key cryptography developed to avoid larger cryptographic key usage. The asymmetric cryptosystem depends on number theory and mathematical elliptic curves (algebraic structure) to generate short, quick, and robust cryptographic keys. RSA is an incumbent public-key algorithm, but its key size is large. The speed of the encryption always depends on the key size: a smaller key length allows faster encryption. To minimize the key size, elliptic curve cryptography has been proposed as a replacement for the RSA algorithm.

The operational key sizes of both algorithms to achieve similar goals are listed below:

ECC Key size	RSA Key Size
160	1024
224	2048
256	3072
384	7680
512	15360

Table 20.3: Comparison of ECC and RSA key size

While RSA uses a key size of 1024 to encrypt the data, ECC provides equal security with a comparatively smaller key size of 160. For high-level computing, RSA uses a key size of 7680 to implement security, whereas ECC can provide the same level of security with a key size of 384.

- **Quantum Cryptography**

  As the world is increasingly adopting online information sharing, cryptosystems are witnessing a sharp increase in security attacks. Since mathematical encryption uses binary digits (0 and 1), it can be easily eavesdropped on or manipulated using various techniques. Hence, quantum cryptography has been introduced to protect data from theft midway (e.g., MITM attacks). This cryptography is processed based on quantum mechanics, such as quantum key distribution (QKD), using photons instead of mathematics as a part of encryption.

  In quantum cryptography, the data are encrypted by a sequence of photons that have a spinning trait while traveling from one end to another end. These photons keep changing their shapes during their course through filters: vertical, horizontal, forward slash, and backslash. Here, vertical and backslash spins imply "ones," while horizontal and forward slash spins imply "zeros."

  - Horizontal (–): 0
  - Vertical (|): 1
  - Backslash (/): 1
  - Forward slash (\): 0

  Attackers can eavesdrop on but cannot manipulate the data because the photons are transferred through arbitrary filters. To breach this mechanism, attackers have to know the exact shape of the photons; if they fail to choose the right transmission, the photon polarization is distorted and the receiver detects an error indicating the eavesdrop.

- **Homomorphic Encryption**

  Homomorphic encryption differs from conventional encryption mechanisms, where math operations are performed to encrypt the plaintext. Homographic encryption allows users to secure and leave their data in an encrypted format even while it is being processed or manipulated. In this technique, encryption and decryption are performed by the same key holder. The homomorphic mechanism enables the user/sender to encrypt the confidential data and out-source it to an enterprise via cloud services to process the given data.

  How homomorphic encryption differs from other encryption mechanisms:

  In private key encryption:

  - Only keyholders can generate and decrypt ciphertexts using similar keys.

  In public key encryption:

  - Only the public keyholder generates the ciphertext and the secret keyholder decrypts the ciphertext.

  In homomorphic encryption:

  - The keyholder can generate the ciphertext and anyone can alter the ciphertext, but only the keyholder again can decrypt the data.

  The reason for using this cryptography is that an untrusted entity can manipulate the data. Hence, this mechanism allows the sender himself/herself to encrypt and decrypt the data, allowing anyone to perform mathematical operations on the ciphertext with respect to the rules applied by the sender.

- **Hardware-Based Encryption**

  Hardware-based encryption is a technique that uses computer hardware for assisting or replacing the software when the data encryption process is being performed. Devices that offer encryption techniques can be considered as hardware-based encryption devices. In the implementation of hardware-based encryption, the cryptography technique workload is transferred to the hardware processors, making the system resources free for performing other functions. These devices can also store encryption keys and other sensitive information in secured areas of RAM or other nonvolatile storage devices such as flash memory.

  Hardware encryption devices reduce instruction sets, where only the authorized code can be executed. These devices do not support third-party software, thereby preventing the execution of any malicious programs. Hardware encryption offers many advantages over software encryption, as it can perform rapid processing of algorithm. It provides tamper-resistant key storage and avoids unauthorized code. Some hardware-based encryption devices are wireless access points, Nitrokey, credit card terminals, and network bulk encryptors.

Types of hardware encryption devices

- **TPM**

  Trusted Platform Module (TPM) is a crypto-processor or a chip that is present in the motherboard. It can securely store the encryption keys and perform many cryptographic operations. TPM offers various features such as authenticating platform integrity, providing full disk encryption capabilities, performing password storage, and providing software license protection.

- **HSM**

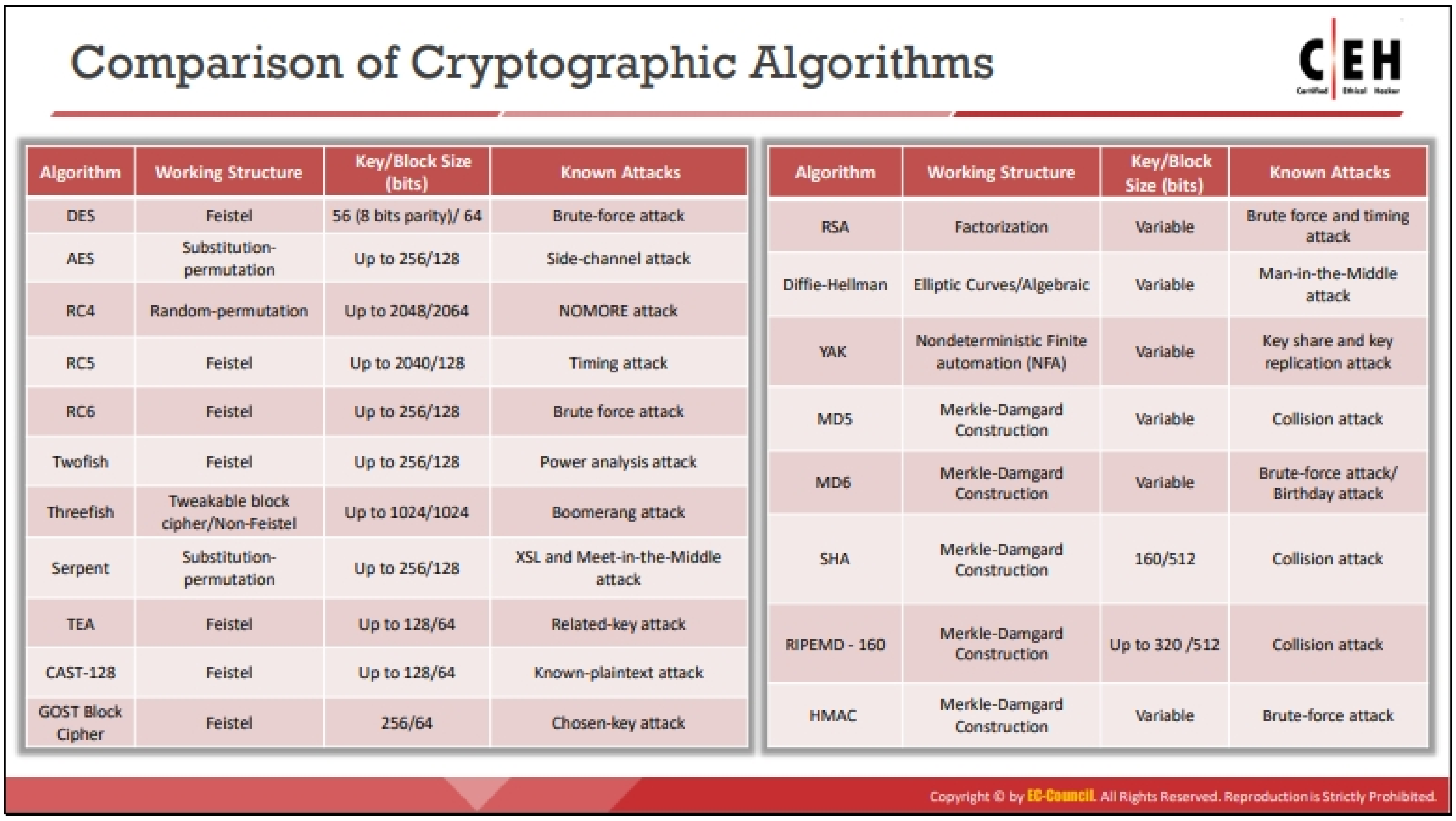
  A hardware security module (HSM) is an additional external security device that is used in a system for crypto-processing, and it can be used for managing, generating, and securely storing cryptographic keys. HSM offers enhanced encryption computation that is useful for symmetric keys longer than 256 bits. High-performance HSM devices are connected to the network using TCP/IP. Some HSM devices include SafeNet Luna Network HSM, nSheild, Cloud HSM, and Cryptosec Dekaton.

- **USB Encryption**

  USB encryption is an additional feature for USB storage devices, which offers onboard encryption services. Encrypted USB devices need an on-device credential system or software- or hardware-based credentials from a computer. USB encryption provides protection against malware distribution over USB and helps in preventing data loss and data leakage. Some hardware USB-encrypted devices include Crypto USB, Kingston Ironkey D300S, and diskAshur Pro 500GB.

- **Hard Drive Encryption**

  Hard drive encryption is a technology whereby the data stored in the hardware can be encrypted using a wide range of encryption options. Hard drive encryption devices cannot use an on-device keyboard or fingerprint reader; instead, they need a TPM or an HSM. These devices can be installed as an internal drive on a computer. Some hard drive encryption devices include military-grade 256-bit AES Hardware Encryption and DiskCypher AES Sata Hard Drive Encryption.



Comparison of Cryptographic Algorithms

Algorithm	Working Structure	Key/Block Size (bits)	Known Attacks
DES	Feistel	56 (8 bits parity)/ 64	Brute-force attack
AES	Substitution-permutation	Up to 256/128	Side-channel attack
RC4	Random-permutation	Up to 2048/2064	NOMORE attack
RC5	Feistel	Up to 2040/128	Timing attack
RC6	Feistel	Up to 256/128	Brute force attack
Twofish	Feistel	Up to 256/128	Power analysis attack
Threefish	Tweakable block cipher/Non-Feistel	Up to 1024/1024	Boomerang attack
Serpent	Substitution-permutation	Up to 256/128	XSL and Meet-in-the-Middle attack
TEA	Feistel	Up to 128/64	Related-key attack
CAST-128	Feistel	Up to 128/64	Known-plaintext attack
GOST Block Cipher	Feistel	256/64	Chosen-key attack

Algorithm	Working Structure	Key/Block Size (bits)	Known Attacks
RSA	Factorization	Variable	Brute force and timing attack
Diffie-Hellman	Elliptic Curves/Algebraic	Variable	Man-in-the-Middle attack
YAK	Nondeterministic Finite automation (NFA)	Variable	Key share and key replication attack
MD5	Merkle-Damgard Construction	Variable	Collision attack
MD6	Merkle-Damgard Construction	Variable	Brute-force attack/ Birthday attack
SHA	Merkle-Damgard Construction	160/512	Collision attack
RIPEMD - 160	Merkle-Damgard Construction	Up to 320 /512	Collision attack
HMAC	Merkle-Damgard Construction	Variable	Brute-force attack

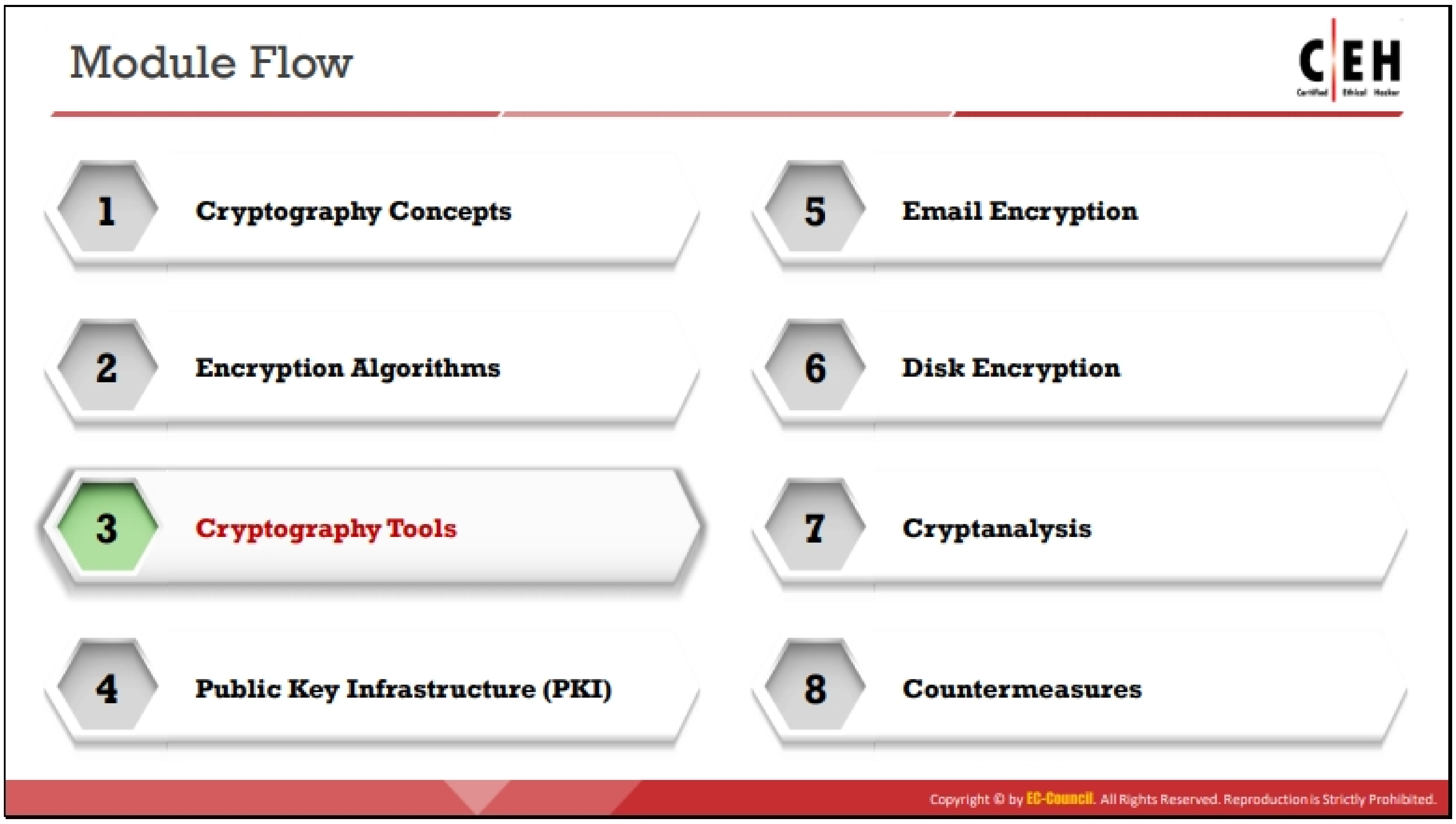
Comparison of Cryptographic Algorithms

The following table compares the various cryptographic algorithms and lists the known attacks against these algorithms:

Algorithm	Working Structure	Key/Block Size (bits)	Known Attacks
DES	Feistel	56 (8 bits parity)/64	Brute-force attack
AES	Substitution-permutation	Up to 256/128	Side-channel attack
RC4	Random-permutation	Up to 2048/2064	NOMORE attack
RC5	Feistel	Up to 2040/128	Timing attack
RC6	Feistel	Up to 256/128	Brute-force attack
Twofish	Feistel	Up to 256/128	Power analysis attack
Threefish	Tweakable block cipher/Non-Feistel	Up to 1024/1024	Boomerang attack
Serpent	Substitution-permutation	Up to 256/128	XSL and Meet-in-the-Middle attack
TEA	Feistel	Up to 128/64	Related-key attack
CAST-128	Feistel	Up to 128/64	Known-plaintext attack

GOST Block Cipher	Feistel	256/64	Chosen-key attack
RSA	Factorization	Variable	Brute-force and timing attack
Diffie–Hellman	Elliptic curves/Algebraic	Variable	Man-in-the-Middle attack
YAK	Nondeterministic finite automation (NFA)	Variable	Key share and key replication attack
MD5	Merkle–Damgard Construction	Variable	Collision attack
MD6	Merkle–Damgard Construction	Variable	Brute-force attack/Birthday attack
SHA	Merkle–Damgard Construction	Variable	Collision attack
RIPEMD - 160	Merkle–Damgard Construction	Up to 320/512	Collision attack
HMAC	Merkle–Damgard Construction	Variable	Brute-force attack

Table 20.4: Comparison of cryptographic algorithms

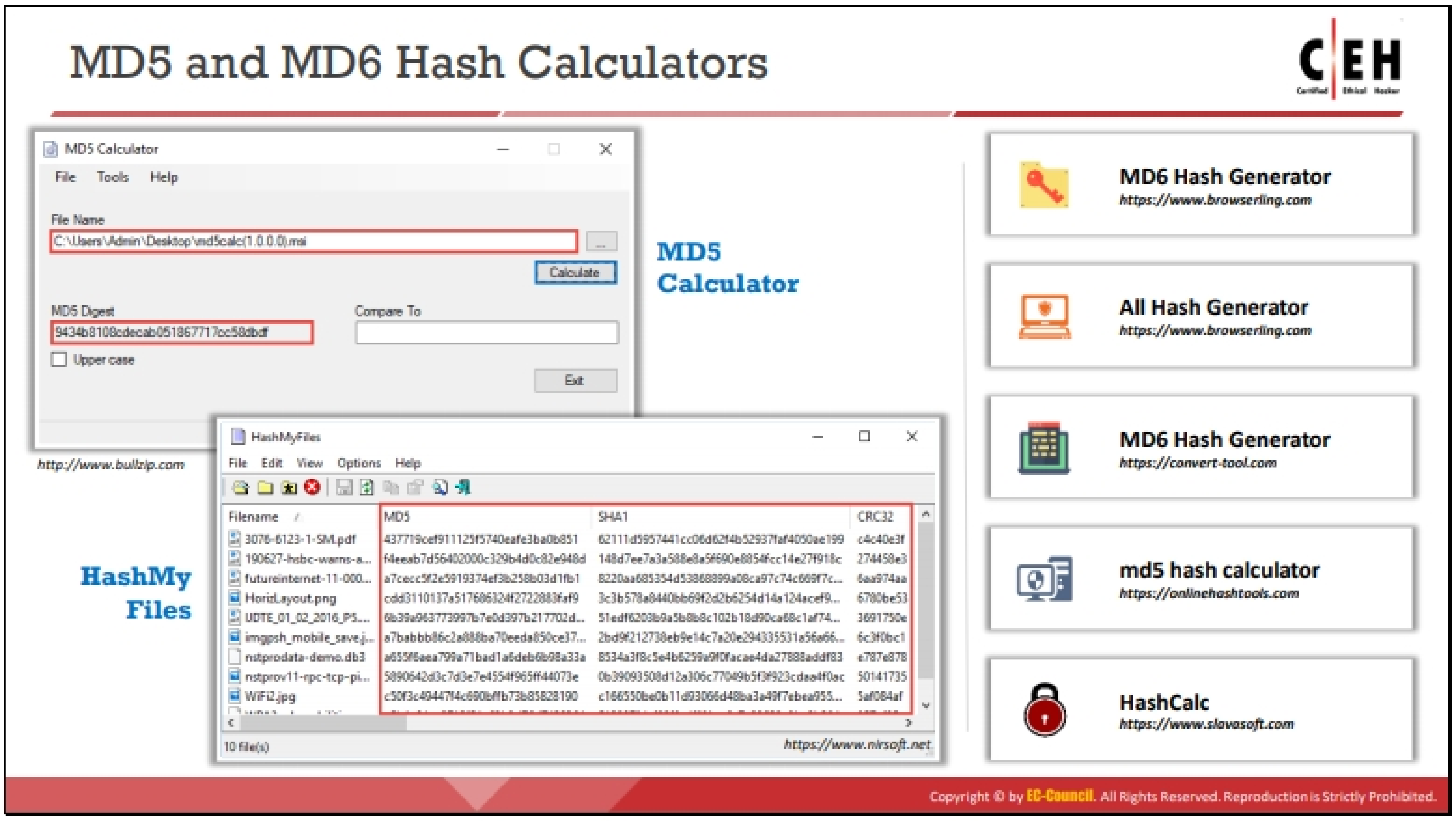


Module Flow

1. Cryptography Concepts
2. Encryption Algorithms
3. Cryptography Tools
4. Public Key Infrastructure (PKI)
5. Email Encryption
6. Disk Encryption
7. Cryptanalysis
8. Countermeasures

Cryptography Tools

This section deals with various cryptography tools that you can use to encrypt sensitive data to protect it from unauthorized access by any party other than the person for whom it is intended.



MD5 and MD6 Hash Calculators

MD5 Calculator

File Name: C:\Users\Admin\Desktop\md5calc(1.0.0.0).msi

MD5 Digest: 9434b8108cdecab051867717cc58dbdf

Compare To:

☐ Upper case

http://www.bullzip.com

HashMy Files

Filename	MD5	SHA1	CRC32
3076-6123-1-SM.pdf	437719cef911125f5740eafe3ba0b851	62111d5957441cc06d62f4b52937faf4050ae199	c4c40e3f
190627-hsbc-warns-a...	f4eeab7d56402000c329b4d0c82e948d	148d7ee7a3a588e8a5f690e8854fcc14e27f918c	274458e3
futureinternet-11-000...	a7cecc5f2e5919374ef3b258b03d1fb1	8220aa685354d53868899a08ca97c74c669f7c...	6aa974aa
HorizLayout.png	cdd3110137a517686324f2722883faf9	3c3b578a8440bb69f2d2b6254d14a124acef9...	6780be53
UDTE_01_02_2016_PS...	6b39a963773997b7e0d397b217702d...	51edf6203b9a5b8b8c102b18d90ca68c1af74...	3691750e
imgpsh_mobile_save.j...	a7babbb86c2a888ba70eeda850ce37...	2bd9f212738eb9e14c7a20e294335531a56a66...	6c3f0bc1
nstprodata-demo.db3	a655f6aea799a71bad1a6deb6b98a33a	8534a3f8c5e4b6259a9f0facae4da27888addf83	e787e878
nstprov11-rpc-tcp-pi...	5890642d3c7d3e7e4554f965ff44073e	0b39093508d12a306c77049b5f3f923cdaa4f0ac	50141735
WiFi2.jpg	c50f3c49447f4c690bffb73b85828190	c166550be0b11d93066d48ba3a49f7ebea955...	5af084af

10 file(s)

https://www.nirsoft.net

MD6 Hash Generator
https://www.browserling.com

All Hash Generator
https://www.browserling.com

MD6 Hash Generator
https://convert-tool.com

md5 hash calculator
https://onlinehashtools.com

HashCalc
https://www.slavasoft.com

MD5 and MD6 Hash Calculators

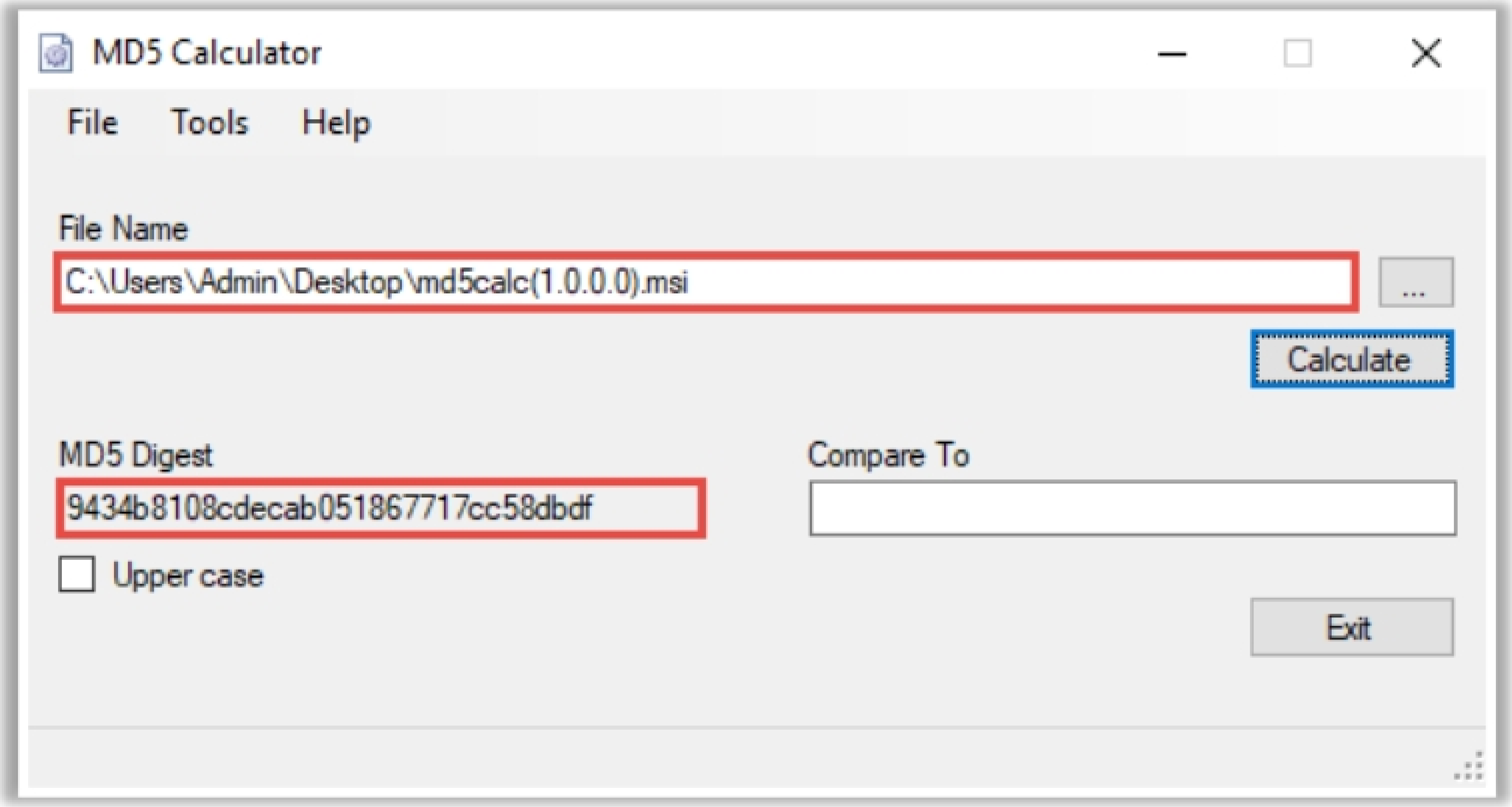
MD5 and MD6 hash calculators that use different hash algorithms to convert plaintext into its equivalent hash value are discussed below.

- **MD5 Calculator**

  Source: *http://www.bullzip.com*

  MD5 Calculator is a simple application that calculates the MD5 hash of a given file. It can be used with large files (e.g., several gigabytes in size). It features a progress counter and a text field from which the final MD5 hash can be easily copied to the clipboard. MD5 Calculator can be used to check the integrity of a file.

  It allows you to calculate the MD5 hash value of the selected file. Right-click the file and choose "**MD5 Calculator**;" the program will calculate the MD5 hash. The MD5 Digest field contains the calculated value. To compare this MD5 digest with another, one can paste the other value into the Compare To field. Obviously, an equal to sign ("=") appears between the two values if they are equal; otherwise, the less than ("<") or greater than (">") sign will tell you that the values are different.



File Name: C:\Users\Admin\Desktop\md5calc(1.0.0.0).msi

MD5 Digest: 9434b8108cdecab051867717cc58dbdf

Compare To:

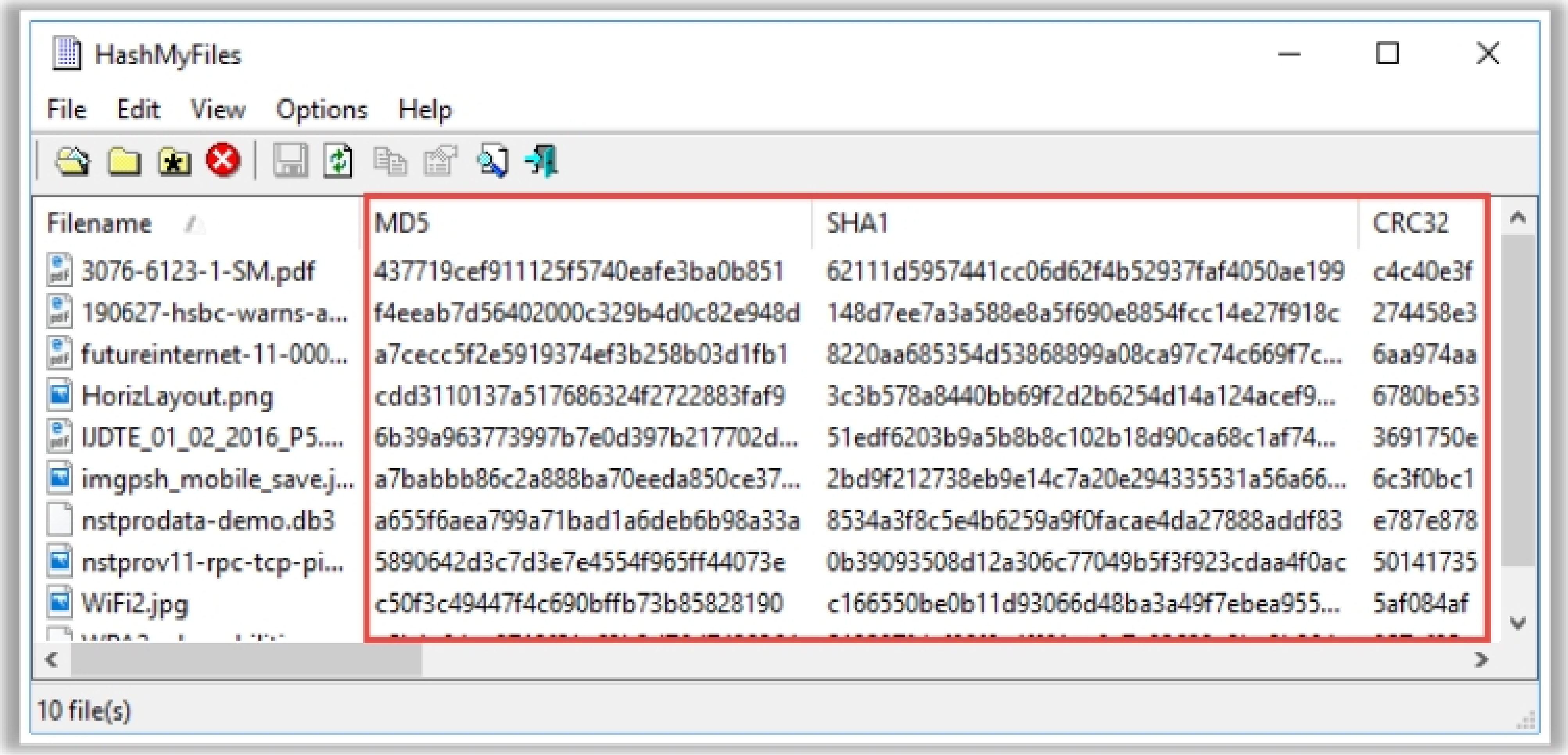
☐ Upper case

Figure 20.10: Screenshot of MD5 Calculator

- **HashMyFiles**

  Source: *https://www.nirsoft.net*

  HashMyFiles is a utility that allows you to calculate the MD5 and SHA1 hashes of one or more files in the system. It allows you to copy the MD5/SHA1 hash list to the clipboard or save it in a text/html/xml file. You can launch HashMyFiles from the context menu of Windows Explorer and display the MD5/SHA1 hashes of the selected files or folders.



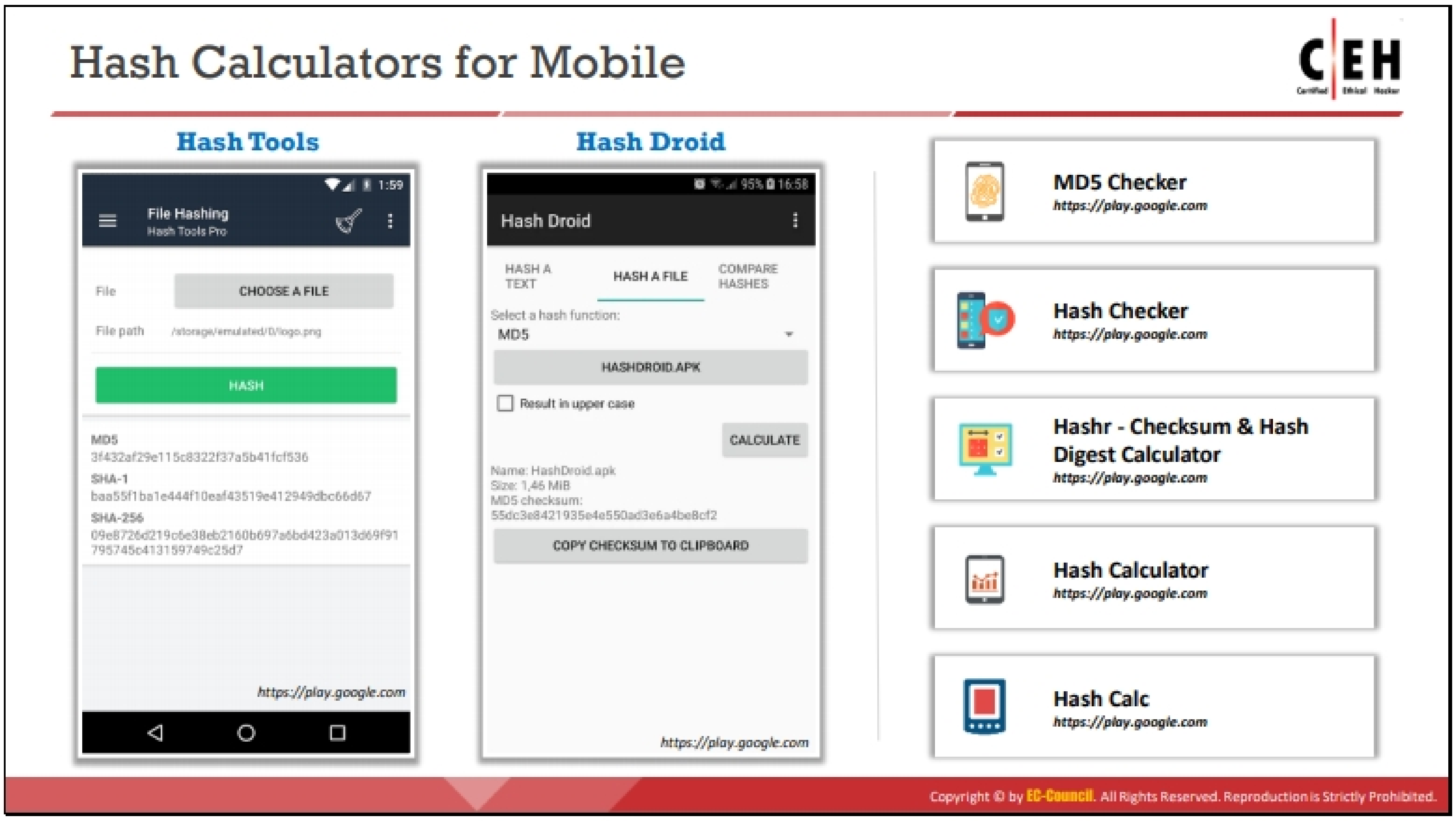
Filename	MD5	SHA1	CRC32
3076-6123-1-SM.pdf	437719cef911125f5740eafe3ba0b851	62111d5957441cc06d62f4b52937faf4050ae199	c4c40e3f
190627-hsbc-warns-a...	f4eeab7d56402000c329b4d0c82e948d	148d7ee7a3a588e8a5f690e8854fcc14e27f918c	274458e3
futureinternet-11-000...	a7cecc5f2e5919374ef3b258b03d1fb1	8220aa685354d53868899a08ca97c74c669f7c...	6aa974aa
HorizLayout.png	cdd3110137a517686324f2722883faf9	3c3b578a8440bb69f2d2b6254d14a124acef9...	6780be53
IJDTE_01_02_2016_P5....	6b39a963773997b7e0d397b217702d...	51edf6203b9a5b8b8c102b18d90ca68c1af74...	3691750e
imgpsh_mobile_save.j...	a7babbb86c2a888ba70eeda850ce37...	2bd9f212738eb9e14c7a20e294335531a56a66...	6c3f0bc1
nstprodata-demo.db3	a655f6aea799a71bad1a6deb6b98a33a	8534a3f8c5e4b6259a9f0facae4da27888addf83	e787e878
nstprov11-rpc-tcp-pi...	5890642d3c7d3e7e4554f965ff44073e	0b39093508d12a306c77049b5f3f923cdaa4f0ac	50141735
WiFi2.jpg	c50f3c49447f4c690bffb73b85828190	c166550be0b11d93066d48ba3a49f7ebea955...	5af084af

10 file(s)

Figure 20.11: Screenshot of HashMyFiles

Some additional MD5 and MD6 hash calculators are as follows:

- MD6 Hash Generator (*https://www.browserling.com*)
- All Hash Generator (*https://www.browserling.com*)
- MD6 Hash Generator (*https://convert-tool.com*)
- md5 hash calculator (*https://onlinehashtools.com*)
- HashCalc (*https://www.slavasoft.com*)



Hash Calculators for Mobile

Hash Tools

1:59
File Hashing
Hash Tools Pro
File
CHOOSE A FILE

File path /storage/emulated/0/logo.png

HASH

MD5
3f432af29e115c8322f37a5b41fcf536
SHA-1
baa55f1ba1e444f10eaf43519e412949dbc66d67
SHA-256
09e8726d219c6e38eb2160b697a6bd423a013d69f91795745c413159749c25d7

https://play.google.com

Hash Droid

95% 16:58
Hash Droid
HASH A TEXT
HASH A FILE
COMPARE HASHES

Select a hash function: MD5

HASHDROID.APK

☐ Result in upper case

CALCULATE

Name: HashDroid.apk
Size: 1,46 MIB
MD5 checksum:
55dc3e8421935e4e550ad3e6a4be8cf2

COPY CHECKSUM TO CLIPBOARD

https://play.google.com

MD5 Checker
https://play.google.com

Hash Checker
https://play.google.com

Hashr - Checksum & Hash Digest Calculator
https://play.google.com

Hash Calculator
https://play.google.com

Hash Calc
https://play.google.com

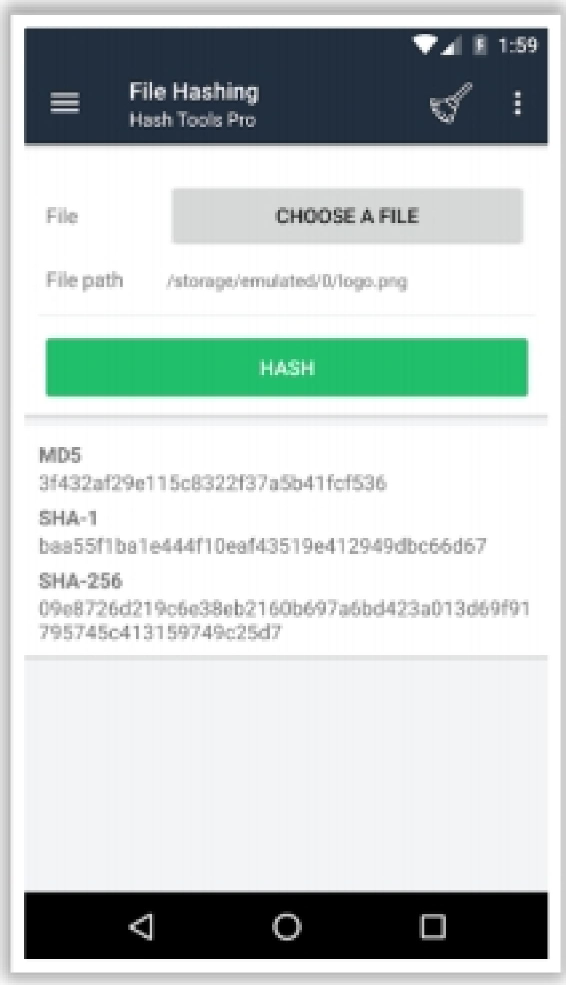
Hash Calculators for Mobile

Some hash calculators for mobile devices are discussed below.

- **Hash Tools**

  Source: *https://play.google.com*

  Hash Tools is a utility for calculating a hash from a given text or decrypting a hash to its original text. In this application, the available hash functions are MD4, MD5, SHA-1, SHA-256, SHA-384, SHA-512, and NTLM.



1:59
File Hashing
Hash Tools Pro
File
CHOOSE A FILE

File path /storage/emulated/0/logo.png

HASH

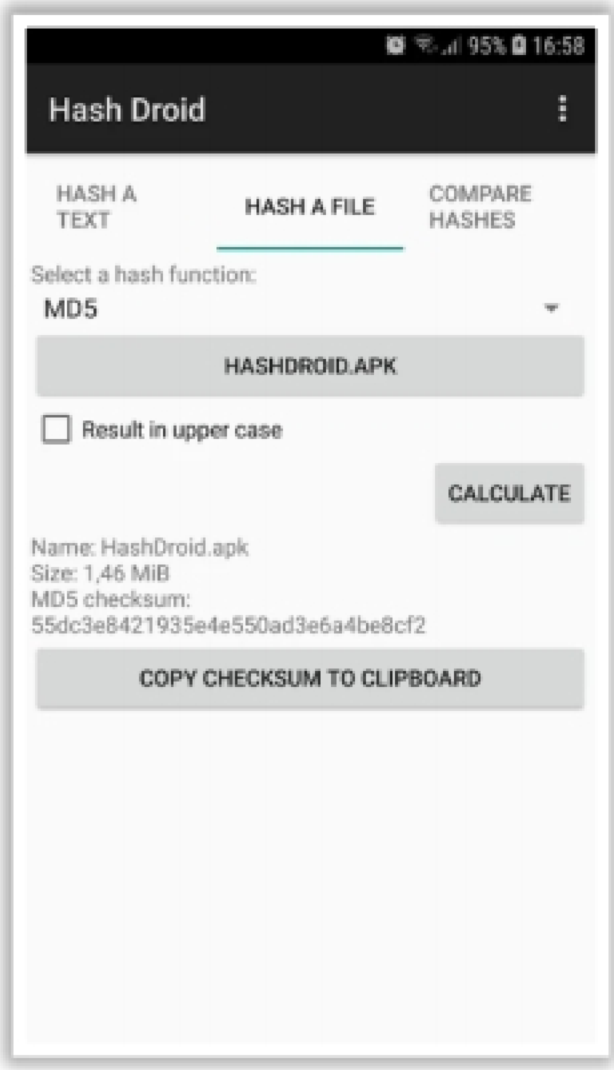
MD5
3f432af29e115c8322f37a5b41fcf536
SHA-1
baa55f1ba1e444f10eaf43519e412949dbc66d67
SHA-256
09e8726d219c6e38eb2160b697a6bd423a013d69f91
795745c413159749c25d7

Figure 20.12: Screenshot of Hash Tools

- **Hash Droid**

  Source: *https://play.google.com*

  The Hash Droid utility helps to calculate a hash from a given text or a file stored on the device. In this application, the available hash functions are Adler-32, CRC-32, Haval-128, MD2, MD4, MD5, RIPEMD-128, RIPEMD-160, SHA-1, SHA-256, SHA-384, SHA-512, Tiger, and Whirlpool.


95% 16:58
Hash Droid
HASH A TEXT
HASH A FILE
COMPARE HASHES
HASHDROID.APK
CALCULATE
COPY CHECKSUM TO CLIPBOARD

Select a hash function:
MD5

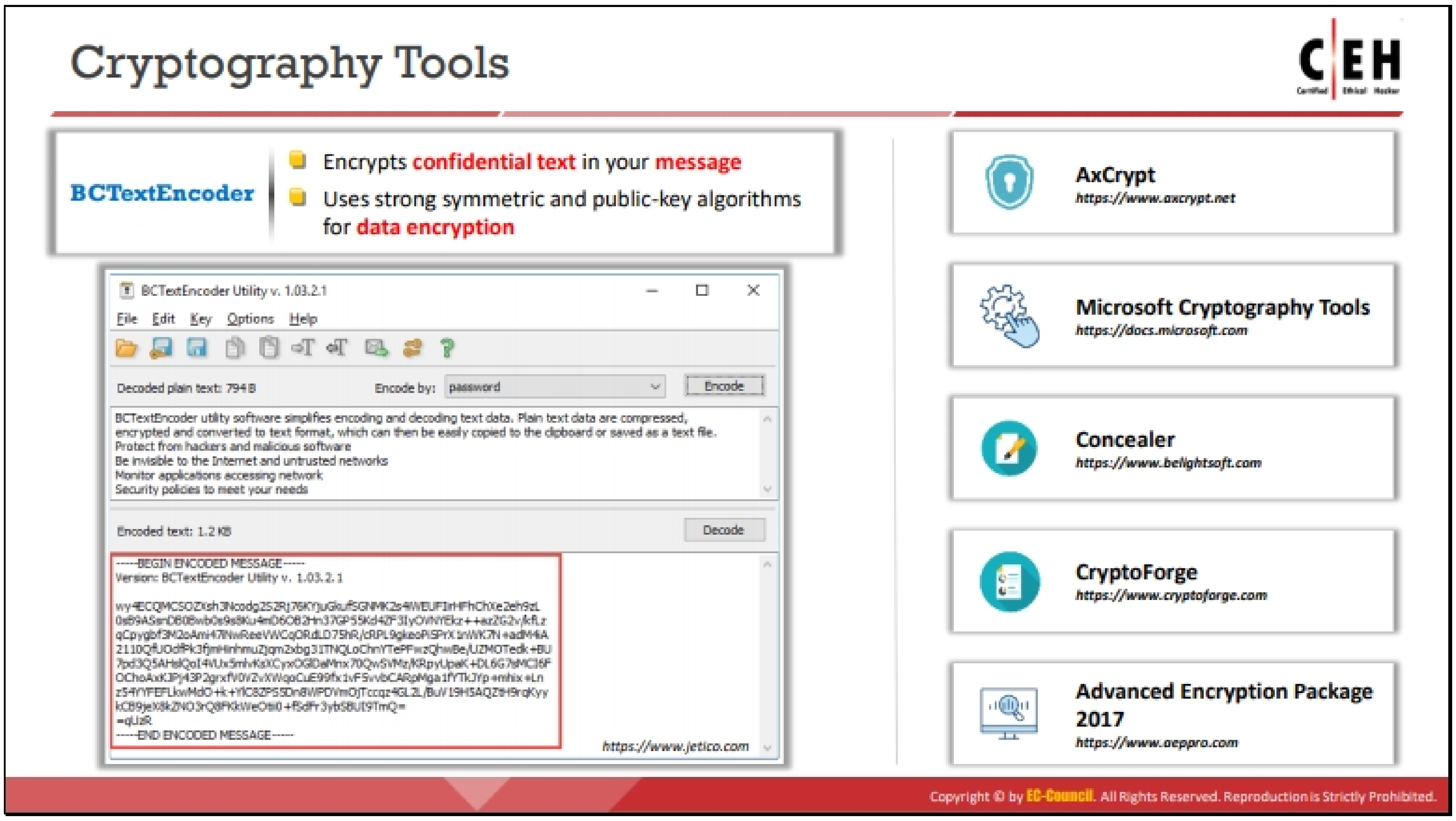
☐ Result in upper case

Name: HashDroid.apk
Size: 1,46 MiB
MD5 checksum:
55dc3e8421935e4e550ad3e6a4be8cf2

Figure 20.13: Screenshot of Hash Droid

Some additional MD5 hash calculators are as follows:

- MD5 Checker (*https://play.google.com*)
- Hash Checker (*https://play.google.com*)
- Hashr - Checksum & Hash Digest Calculator (*https://play.google.com*)
- Hash Calculator (*https://play.google.com*)
- Hash Calc (*https://play.google.com*)



Cryptography Tools

BCTextEncoder

- Encrypts confidential text in your message
- Uses strong symmetric and public-key algorithms for data encryption

BCTextEncoder Utility v. 1.03.2.1

Decoded plain text: 794 B

Encode by: password

```
BCTextEncoder utility software simplifies encoding and decoding text data. Plain text data are compressed,
encrypted and converted to text format, which can then be easily copied to the clipboard or saved as a text file.
Protect from hackers and malicious software
Be invisible to the Internet and untrusted networks
Monitor applications accessing network
Security policies to meet your needs
```

Encoded text: 1.2 KB

```
-----BEGIN ENCODED MESSAGE-----
Version: BCTextEncoder Utility v. 1.03.2.1
```

```
-----END ENCODED MESSAGE-----
```

https://www.jetico.com

AxCrypt
https://www.axcrypt.net

Microsoft Cryptography Tools
https://docs.microsoft.com

Concealer
https://www.belightsoft.com

CryptoForge
https://www.cryptoforge.com

Advanced Encryption Package 2017
https://www.aeppro.com

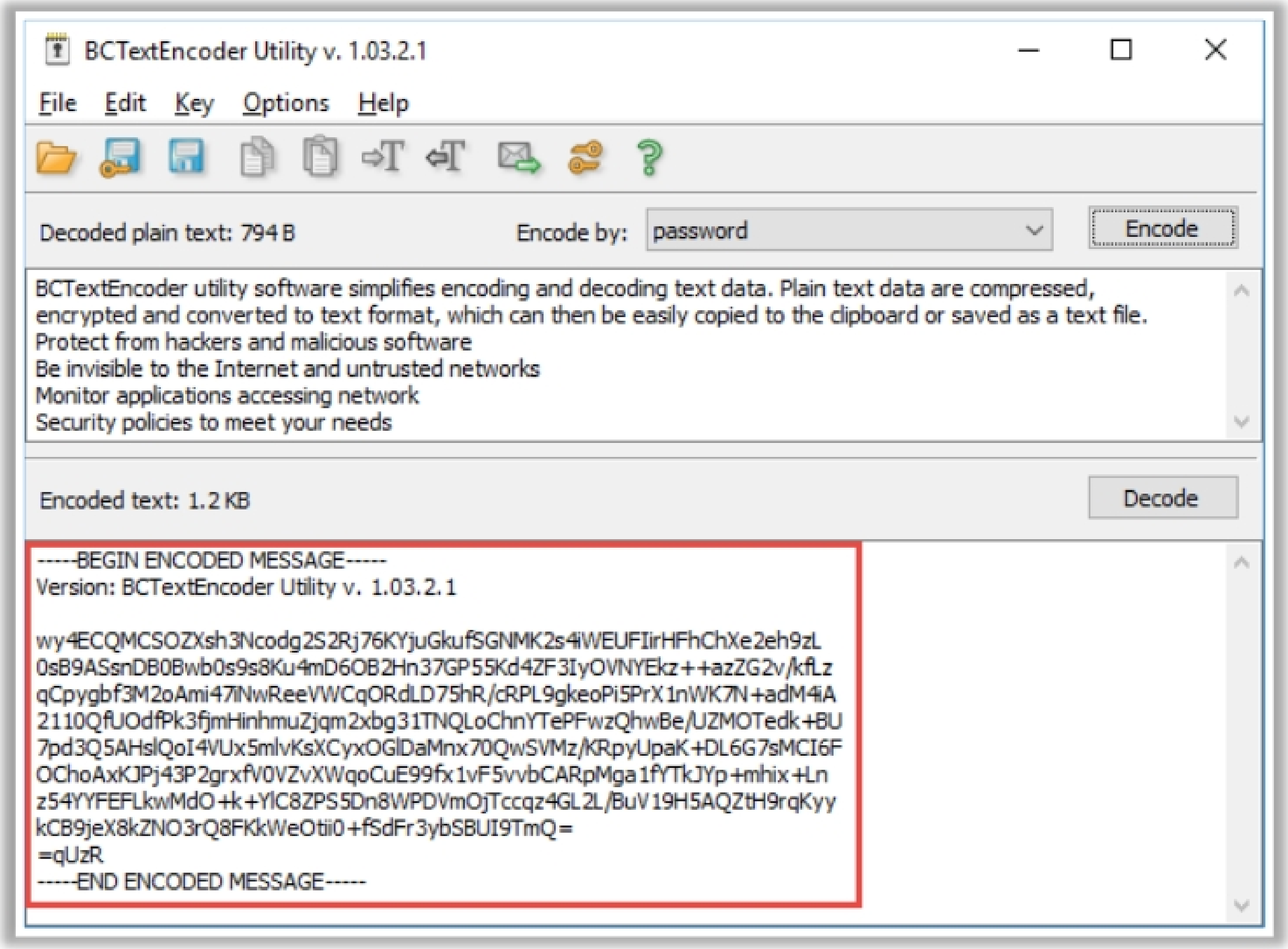
Cryptography Tools

You can use various cryptographic tools for encrypting and decrypting your information, files, etc. These tools implement different types of encryption algorithms.

- **BCTextEncoder**

  Source: *https://www.jetico.com*

  The BCTextEncoder utility simplifies the encoding and decoding of text data. It compresses, encrypts, and converts plaintext data into text format, which the user can then copy to the clipboard or save as a text file. It uses public key encryption methods as well as password-based encryption. Furthermore, it uses strong and approved symmetric and public-key algorithms for data encryption.



Decoded plain text: 794 B

Encode by: password

BCTextEncoder utility software simplifies encoding and decoding text data. Plain text data are compressed, encrypted and converted to text format, which can then be easily copied to the clipboard or saved as a text file.
Protect from hackers and malicious software
Be invisible to the Internet and untrusted networks
Monitor applications accessing network
Security policies to meet your needs

Encoded text: 1.2 KB

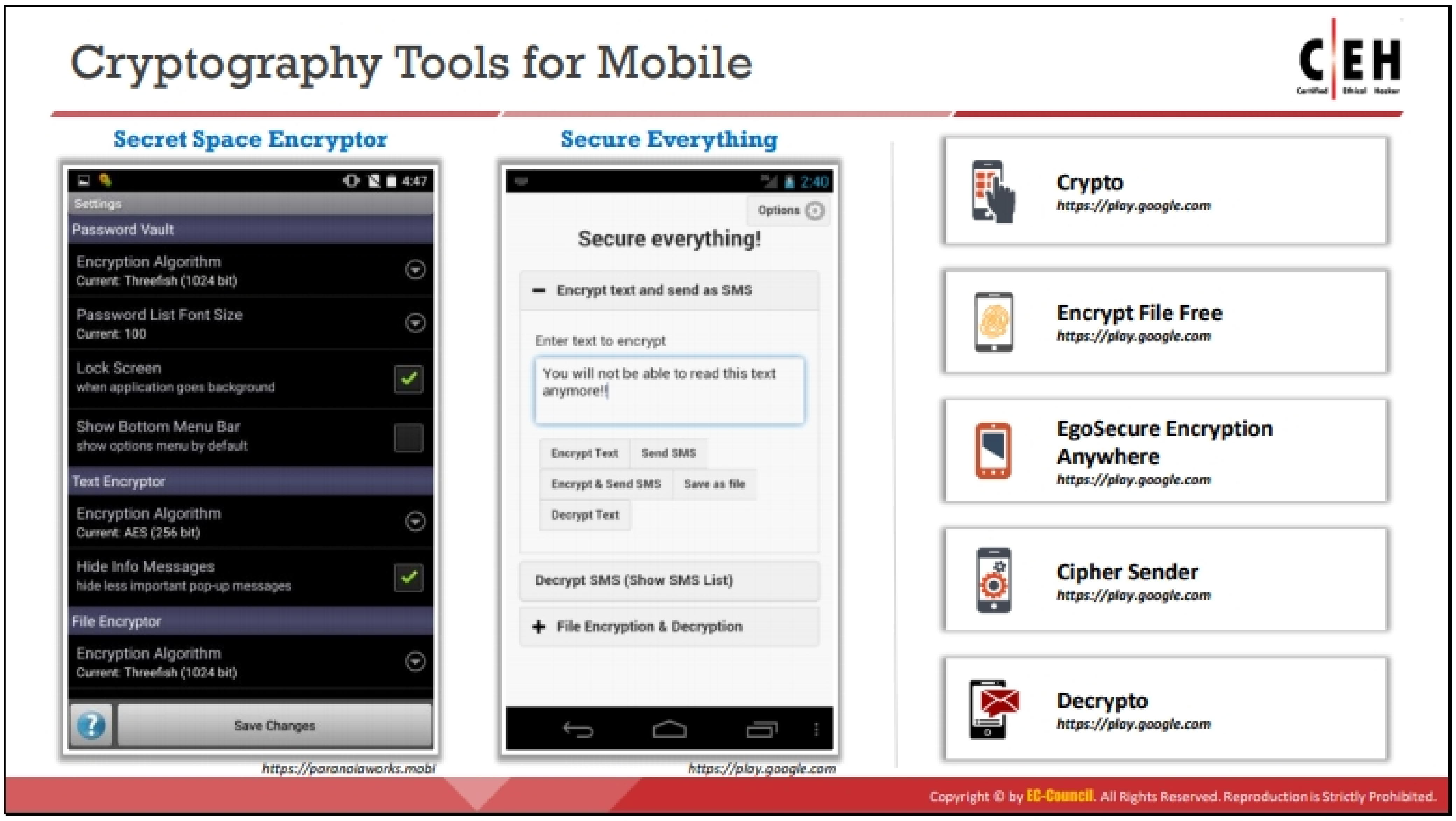
```
-----BEGIN ENCODED MESSAGE-----
Version: BCTextEncoder Utility v. 1.03.2.1

wy4ECQMCSOZXsh3Ncodg2S2Rj76KYjuGkufSGNMK2s4iWEUFIirHFhChXe2eh9zL
0sB9ASsnDB0Bwb0s9s8Ku4mD6OB2Hn37GP55Kd4ZF3IyOVNYEkz++azZG2v/kfLz
qCpygbf3M2oAmi47lNwReeVWCqORdLD75hR/cRPL9gkeoPi5PrX1nWK7N+adM4iA
2110QfUOdfPk3fjmHinhmuZjqm2xbg31TNQLoChnYTePFwzQhwBe/UZMOTedk+BU
7pd3Q5AHslQoI4VUx5mlvKsXCyxOGlDaMnx70QwSVMz/KRpyUpaK+DL6G7sMCI6F
OChoAxKJPj43P2grxfV0VZvXWqoCuE99fx1vF5vvbCARpMga1fYTkJYp+mhix+Ln
z54YYFEFLkwMdO+k+YlC8ZPS5Dn8WPDVmOjTccqz4GL2L/BuV19H5AQZtH9rqKyy
kCB9jeX8kZNO3rQ8FKkWeOtii0+fSdFr3ybSBUI9TmQ=
=qUzR
-----END ENCODED MESSAGE-----
```

Figure 20.14: Screenshot of BCTextEncoder

Some additional cryptography tools are as follows:

- AxCrypt (*https://www.axcrypt.net*)
- Microsoft Cryptography Tools (*https://docs.microsoft.com*)
- Concealer (*https://www.belightsoft.com*)
- SensiGuard (*https://www.sensiguard.com*)
- Challenger (*https://www.encryption-software.de*)



Cryptography Tools for Mobile

Secret Space Encryptor

Settings
Password Vault
Encryption Algorithm — Current: Threefish (1024 bit)
Password List Font Size — Current: 100
Lock Screen — when application goes background ☑
Show Bottom Menu Bar — show options menu by default ☐
Text Encryptor
Encryption Algorithm — Current: AES (256 bit)
Hide Info Messages — hide less important pop-up messages ☑
File Encryptor
Encryption Algorithm — Current: Threefish (1024 bit)

https://paranoiaworks.mobi

Secure Everything

Secure everything!
Encrypt text and send as SMS
Enter text to encrypt
You will not be able to read this text anymore!!
Decrypt SMS (Show SMS List)
File Encryption & Decryption

https://play.google.com

Crypto
https://play.google.com

Encrypt File Free
https://play.google.com

EgoSecure Encryption Anywhere
https://play.google.com

Cipher Sender
https://play.google.com

Decrypto
https://play.google.com

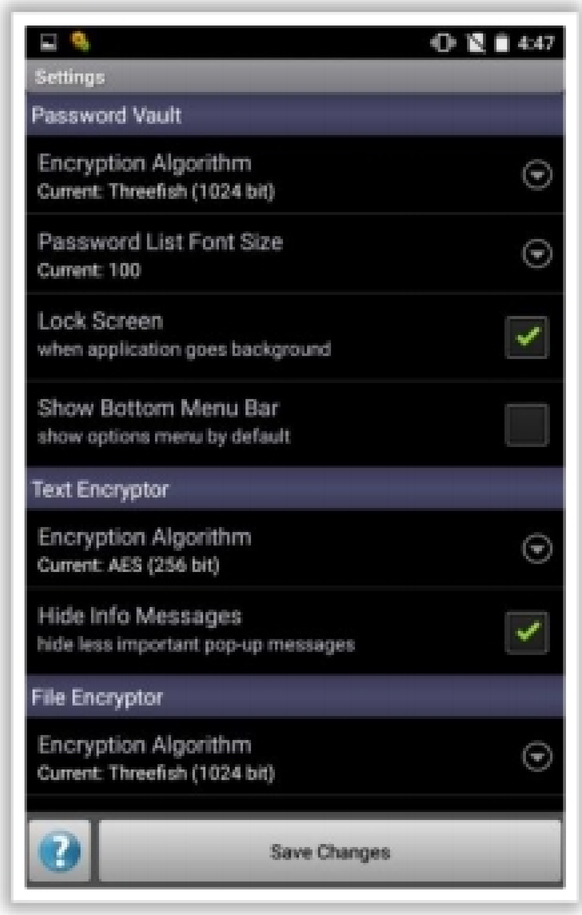
Cryptography Tools for Mobile

Some cryptographic tools for mobile devices are discussed below:

- **Secret Space Encryptor**

  Source: *https://paranoiaworks.mobi*

  Secret Space Encryptor is an integrated solution for password management, message (text) encryption, and file encryption. It keeps messages, notes, and other text safe from unintended readers. It uses encryption algorithms such as AES (Rijndael) 256bit, RC6 256bit, Serpent 256bit, Blowfish 256bit/448bit, Twofish 256bit, and GOST 256bit.



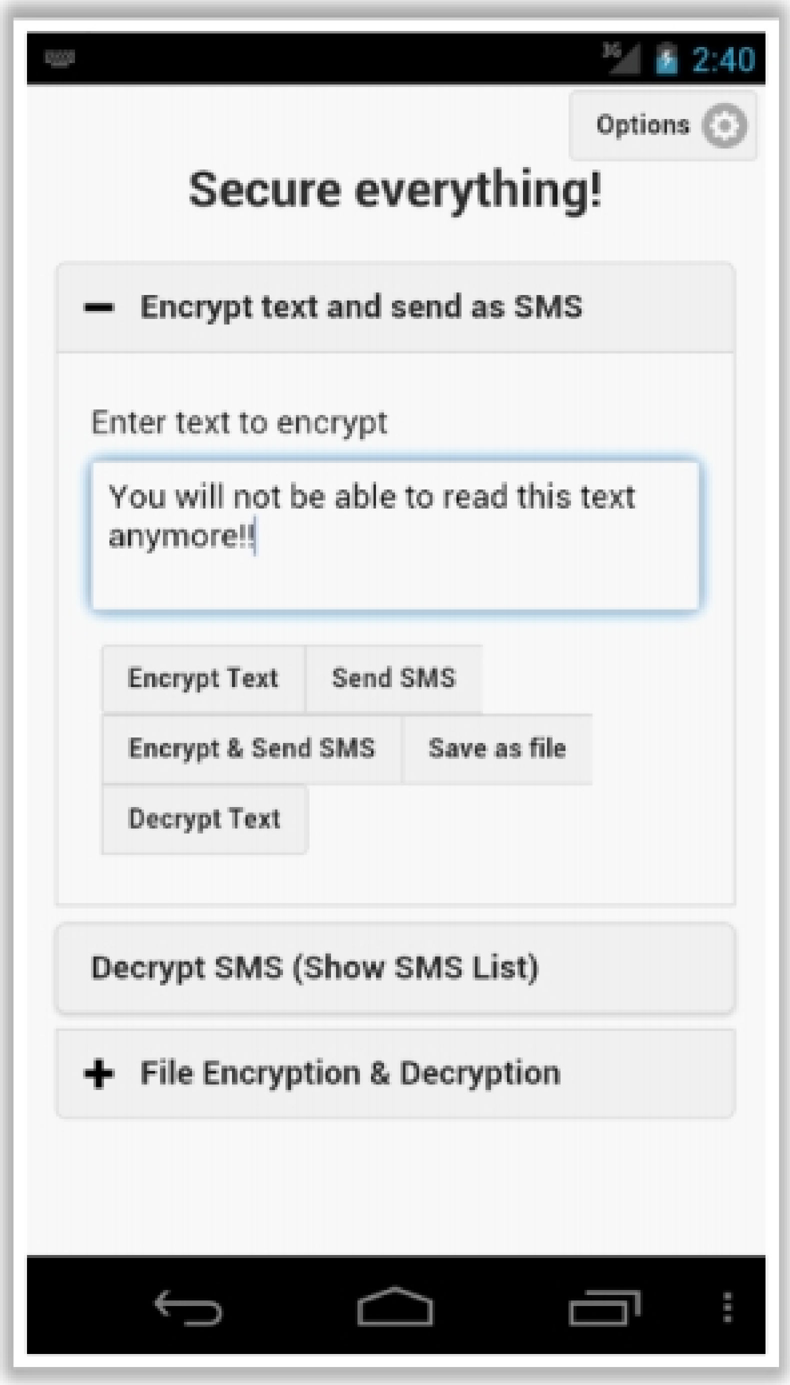
Settings
Password Vault
Encryption Algorithm — Current: Threefish (1024 bit)
Password List Font Size — Current: 100
Lock Screen — when application goes background ☑
Show Bottom Menu Bar — show options menu by default ☐
Text Encryptor
Encryption Algorithm — Current: AES (256 bit)
Hide Info Messages — hide less important pop-up messages ☑
File Encryptor
Encryption Algorithm — Current: Threefish (1024 bit)

Figure 20.15: Screenshot of Secret Space Encryptor

- **Secure Everything**

  Source: *https://play.google.com*

  Secure Everything uses AES encryption to secure SMS, videos, images, audio files, etc. This tool also helps in securing credit card details, bank account details, SSN, etc.



2:40
Options
Secure everything!
Encrypt text and send as SMS
Encrypt Text
Send SMS
Encrypt & Send SMS
Save as file
Decrypt Text
Decrypt SMS (Show SMS List)
File Encryption & Decryption

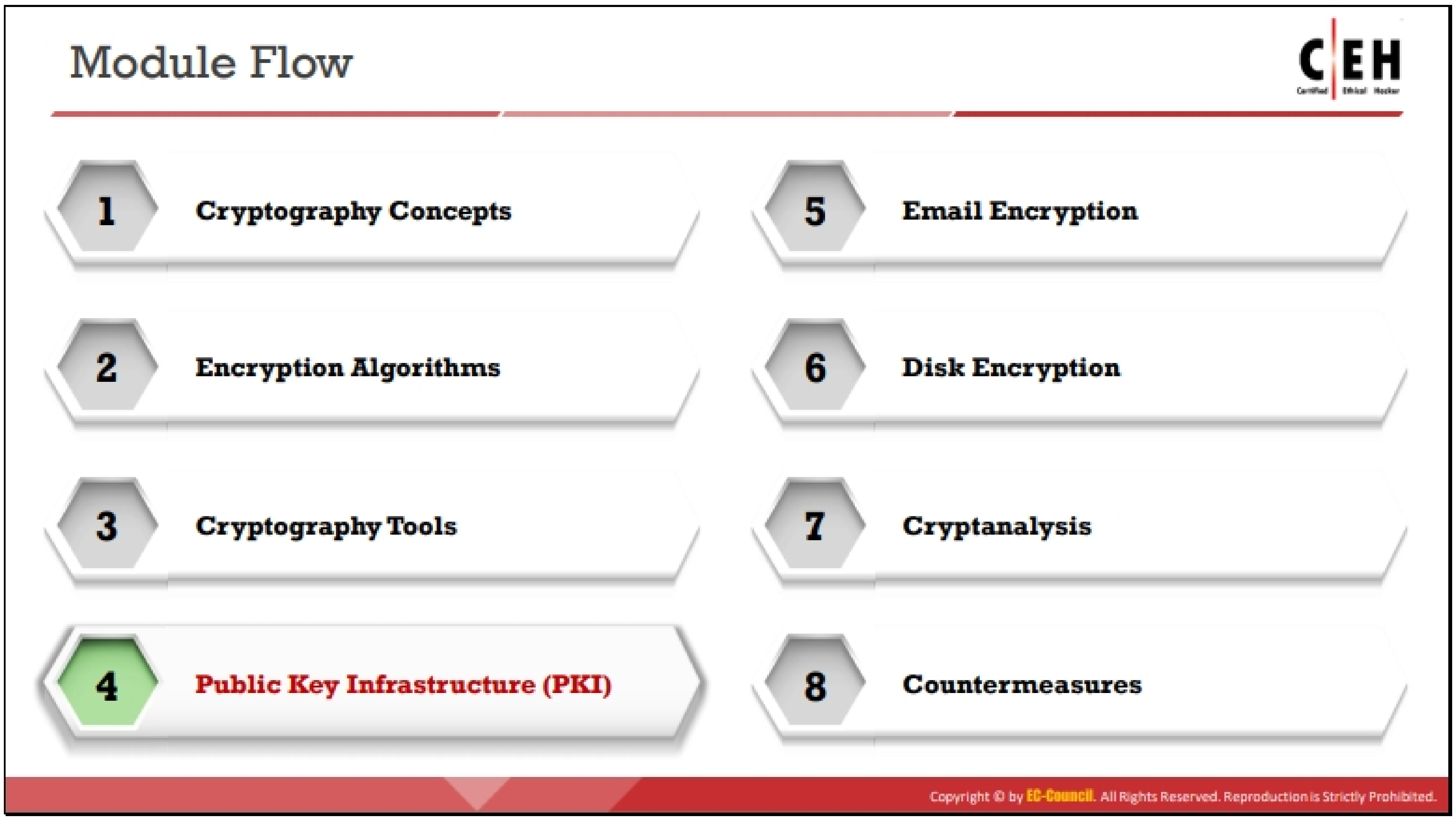
Enter text to encrypt

You will not be able to read this text anymore!!

Figure 20.16: Screenshot of Secure Everything

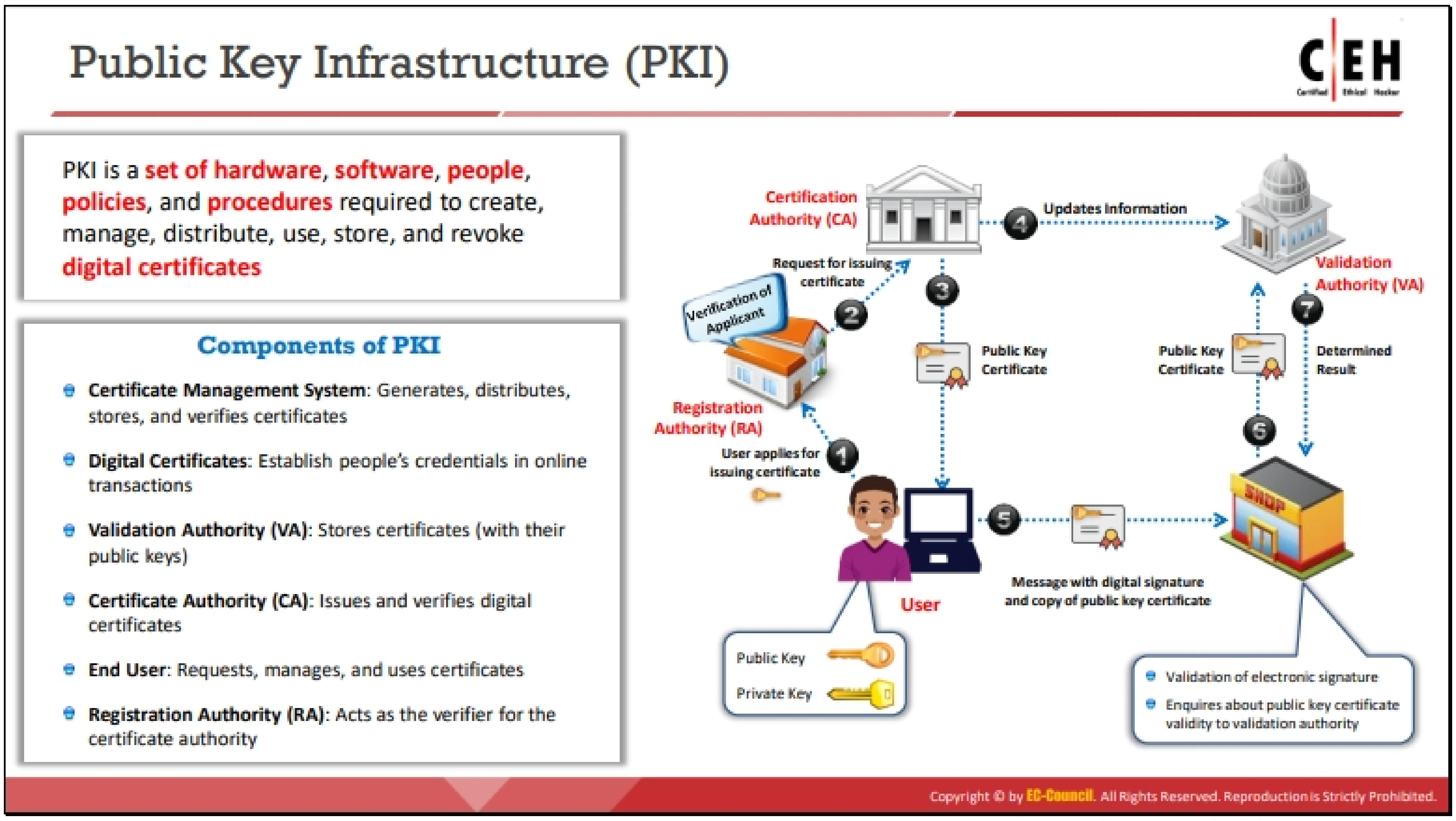
Some additional cryptography tools for mobile devices are as follows:

- Crypto (*https://play.google.com*)
- Encrypt File Free (*https://play.google.com*)
- EgoSecure Encryption Anywhere (*https://play.google.com*)
- Cipher Sender (*https://play.google.com*)
- Decrypto (*https://play.google.com*)



Module Flow

1. **Cryptography Concepts**
2. **Encryption Algorithms**
3. **Cryptography Tools**
4. **Public Key Infrastructure (PKI)**
5. **Email Encryption**
6. **Disk Encryption**
7. **Cryptanalysis**
8. **Countermeasures**



Public Key Infrastructure (PKI)

PKI is a **set of hardware**, **software**, **people**, **policies**, and **procedures** required to create, manage, distribute, use, store, and revoke **digital certificates**

Components of PKI

- **Certificate Management System**: Generates, distributes, stores, and verifies certificates
- **Digital Certificates**: Establish people's credentials in online transactions
- **Validation Authority (VA)**: Stores certificates (with their public keys)
- **Certificate Authority (CA)**: Issues and verifies digital certificates
- **End User**: Requests, manages, and uses certificates
- **Registration Authority (RA)**: Acts as the verifier for the certificate authority

Certification Authority (CA)
4 Updates Information
Validation Authority (VA)
Request for issuing certificate
Verification of Applicant
2
3
Public Key Certificate
Public Key Certificate
7
Determined Result
Registration Authority (RA)
6
User applies for issuing certificate
1
5
User
Message with digital signature and copy of public key certificate
SHOP
Public Key
Private Key
Validation of electronic signature
Enquires about public key certificate validity to validation authority

Public Key Infrastructure (PKI)

This section deals with public key infrastructure (PKI) and the role of each component of PKI, certification authorities such as Comodo, IdenTrust, Symantec, and GoDaddy, and signed certificates (CA) vs. the self-signed certificates.

PKI is a security architecture developed to increase the confidentiality of information exchanged over the insecure Internet. It includes hardware, software, people, policies, and procedures required to create, manage, distribute, use, store, and revoke digital certificates. In cryptography, the PKI helps to bind public keys with corresponding user identities by means of a certification authority (CA).

Components of PKI

- **Certificate Management System**: Generates, distributes, stores, and verifies certificates
- **Digital Certificates**: Establishes credentials of a person when performing online transactions
- **Validation Authority (VA)**: Stores certificates (with their public keys)
- **Certification Authority (CA)**: Issues and verifies digital certificates
- **End User**: Requests, manages, and uses certificates
- **Registration Authority (RA)**: Acts as the verifier for the CA

PKI is a comprehensive system that allows the use of public-key encryption and digital signature services across a wide variety of applications. PKI authentication depends on digital certificates (also known as public-key certificates) that CAs sign and provide. A digital certificate is a digitally signed statement with a public key and the subject (user, company, or system) name in it.

PKI uses public-key cryptography, which is widely used on the Internet to encrypt messages or authenticate message senders. In public-key cryptography, a CA generates public and private keys with the same algorithm simultaneously. The private key is held only by the subject (user, company, or system) mentioned in the certificate, while the public key is made publicly available in a directory that all parties can access. The subject keeps the private key secret and uses it to decrypt the text encrypted by someone else using the corresponding public key (available in a public directory). Thus, others encrypt messages for the user with the user's public key, and the user decrypts it with his/her private key.

The steps involved in the PKI process are as follows:

1. The subject (user, company, or system) intending to exchange information securely applies for a certificate to the registration authority (RA).
2. The RA receives the request from the subject, verifies the subject's identity, and requests the CA to issue a public key certificate to the user.
3. The CA issues the public key certificate binding the subject's identity with the subject's public key; then, the updated information is sent to the validation authority (VA).
4. When a user makes a transaction, the user duly signs the message digitally using the public key certificate and sends the message to the client.
5. The client verifies the authenticity of the user by inquiring with the VA about the validity of the user's public key certificate.
6. The VA compares the public key certificate of the user with that of the updated information provided by the CA and determines the result (valid or invalid).

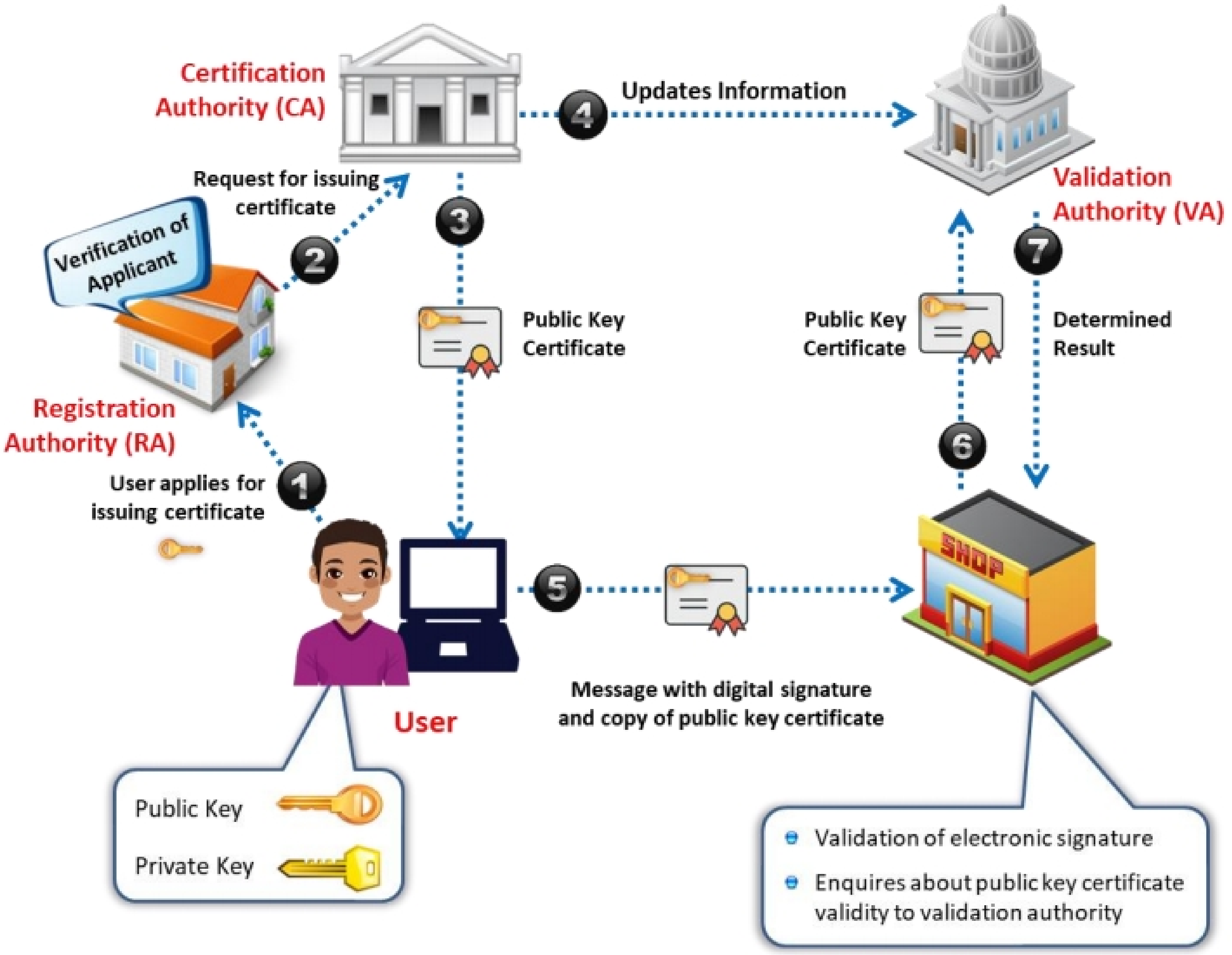
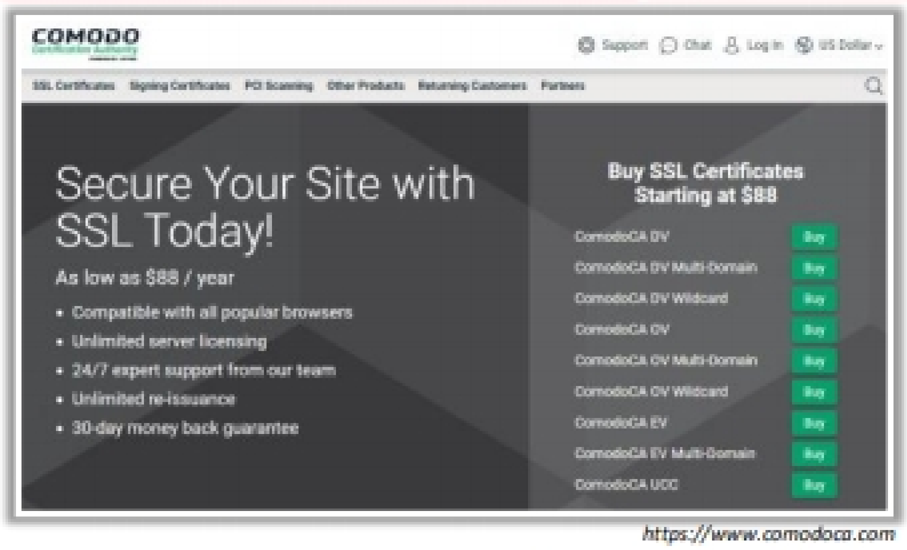

Certification Authority (CA)
4
Updates Information
Validation Authority (VA)
Request for issuing certificate
3
Verification of Applicant
2
7
Public Key Certificate
Public Key Certificate
Determined Result
Registration Authority (RA)
6
User applies for issuing certificate
1
SHOP
5
User
Message with digital signature and copy of public key certificate
Public Key
Private Key
Validation of electronic signature
Enquires about public key certificate validity to validation authority

Figure 20.17: Public Key Infrastructure (PKI)

Certification Authorities


COMODO
Support
Chat
Log In
US Dollar
SSL Certificates
Signing Certificates
PCI Scanning
Other Products
Returning Customers
Partners
Buy

Secure Your Site with SSL Today!

As low as $88 / year

- Compatible with all popular browsers
- Unlimited server licensing
- 24/7 expert support from our team
- Unlimited re-issuance
- 30-day money back guarantee

Buy SSL Certificates Starting at $88

ComodoCA DV
ComodoCA DV Multi-Domain
ComodoCA DV Wildcard
ComodoCA OV
ComodoCA OV Multi-Domain
ComodoCA OV Wildcard
ComodoCA EV
ComodoCA EV Multi-Domain
ComodoCA UCC

https://www.comodoca.com

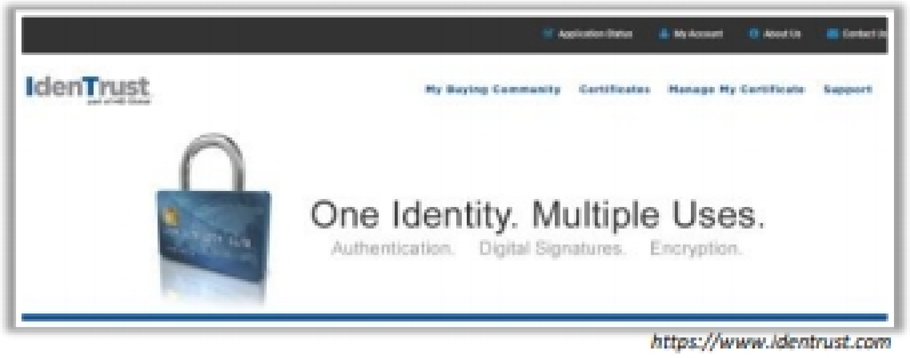
GoDaddy
Help by GoDaddy Guides
Sign In

SSL Certificates

Get an SSL certificate. Show visitors you're trustworthy and authentic.

Call for a free security assessment: (480) 463-8327

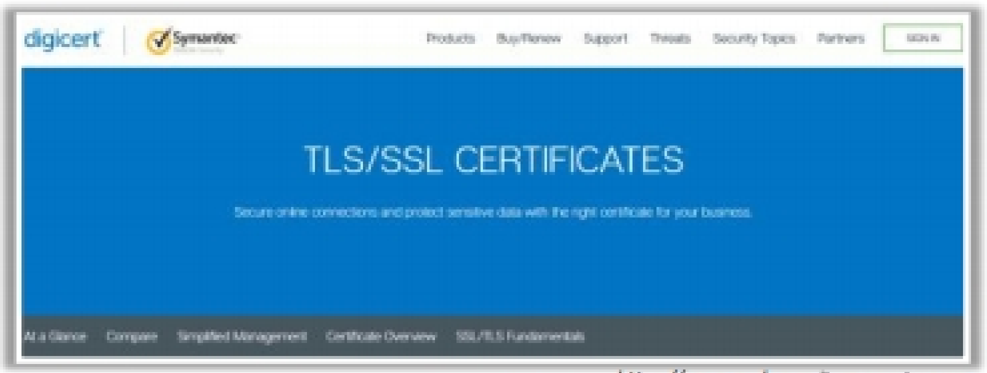
https://www.godaddy.com


Application Status
My Account
About Us
Contact Us
IdenTrust
My Buying Community
Certificates
Manage My Certificate
Support

One Identity. Multiple Uses.

Authentication. Digital Signatures. Encryption.

https://www.identrust.com


digicert
Symantec
Products
Buy/Renew
Support
Threats
Security Topics
Partners
SIGN IN
At a Glance
Compare
Simplified Management
Certificate Overview
SSL/TLS Fundamentals

TLS/SSL CERTIFICATES

Secure online connections and protect sensitive data with the right certificate for your business.

https://www.websecurity.symantec.com

Certification Authorities

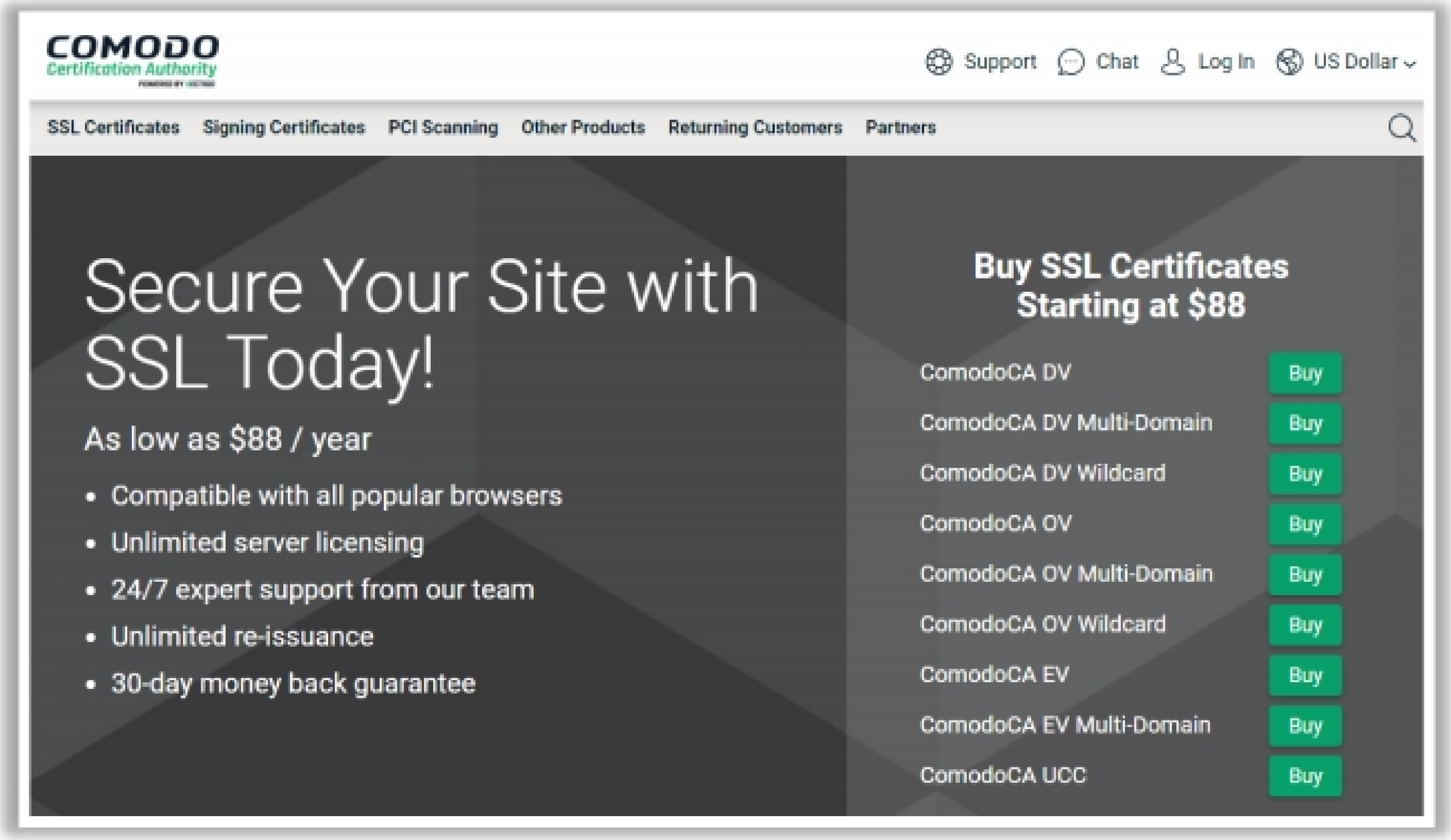
Certification authorities (CAs) are trusted entities that issue digital certificates. The digital certificate certifies the possession of the public key by the subject (user, company, or system) specified in the certificate. This aids others to trust signatures or statements made by the private key that is associated with the certified public key.

Some popular CAs are discussed below:

- **Comodo**

  Source: *https://www.comodoca.com*

  Comodo offers a range of PKI digital certificates with strong SSL encryption (128/256 available) with Server-Gated Cryptography (SGC). It ensures standards of confidentiality, system reliability, and pertinent business practices as judged via qualified independent audits. It offers PKI management solutions such as Comodo Certificate Manager and Comodo EPKI Manager.


COMODO
Certification Authority
Support
Chat
Log In
US Dollar
SSL Certificates
Signing Certificates
PCI Scanning
Other Products
Returning Customers
Partners
Buy

Secure Your Site with SSL Today!

As low as $88 / year

- Compatible with all popular browsers
- Unlimited server licensing
- 24/7 expert support from our team
- Unlimited re-issuance
- 30-day money back guarantee

Buy SSL Certificates Starting at $88

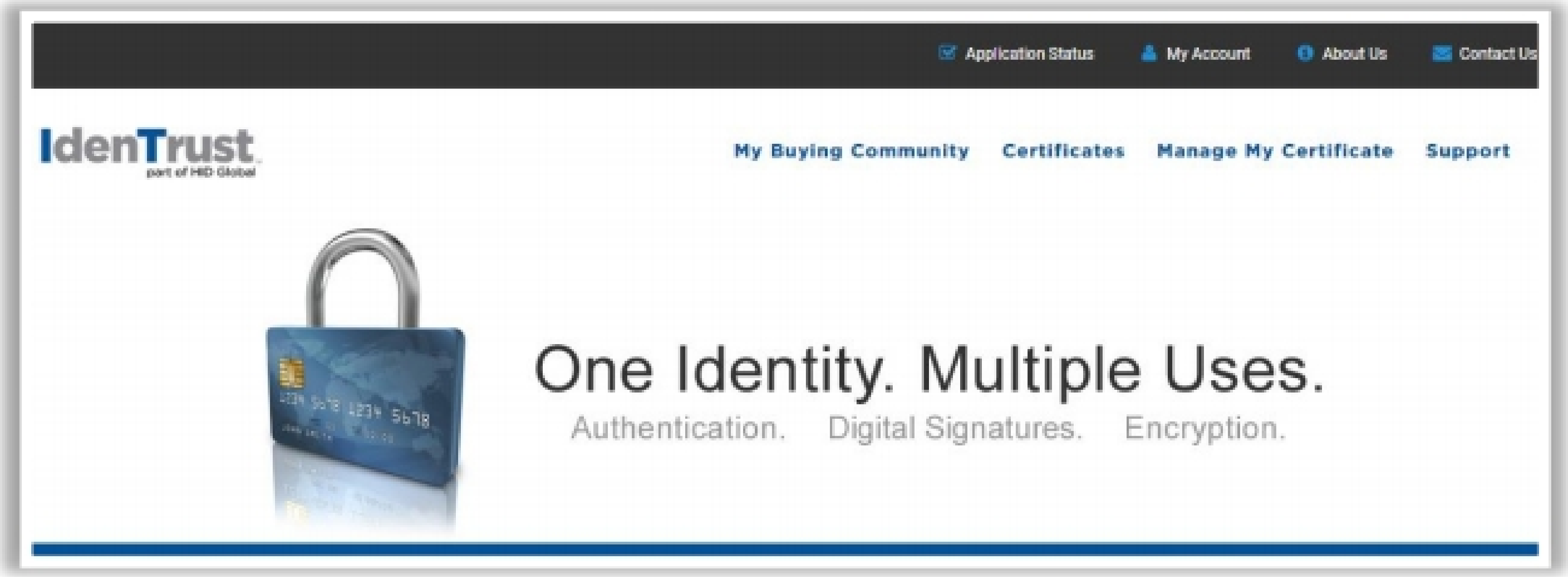
- ComodoCA DV
- ComodoCA DV Multi-Domain
- ComodoCA DV Wildcard
- ComodoCA OV
- ComodoCA OV Multi-Domain
- ComodoCA OV Wildcard
- ComodoCA EV
- ComodoCA EV Multi-Domain
- ComodoCA UCC

Figure 20.18: Screenshot of Comodo Website

- **IdenTrust**

  Source: *https://www.identrust.com*

  IdenTrust is a trusted third party that provides CA services for many sectors such as banks, corporates, governments, and healthcare. It provides solutions such as digital signing and sealing, compliance with NIST SP 800-171, global identity networks, and managed PKI hosting services.


Application Status
My Account
About Us
Contact Us
IdenTrust
part of HID Global
My Buying Community
Certificates
Manage My Certificate
Support

One Identity. Multiple Uses.
Authentication. Digital Signatures. Encryption.

Figure 20.19: Screenshot of IdenTrust Website

- **Symantec**

  Source: *https://www.websecurity.symantec.com*

  Symantec Corporation (NASDAQ: SYMC) provides solutions that allow companies and consumers to engage in communications and commerce online with confidence. Symantec offers SSL/TLS certificates such as Secure Site, Secure Site with EV, and Secure Site Pro.

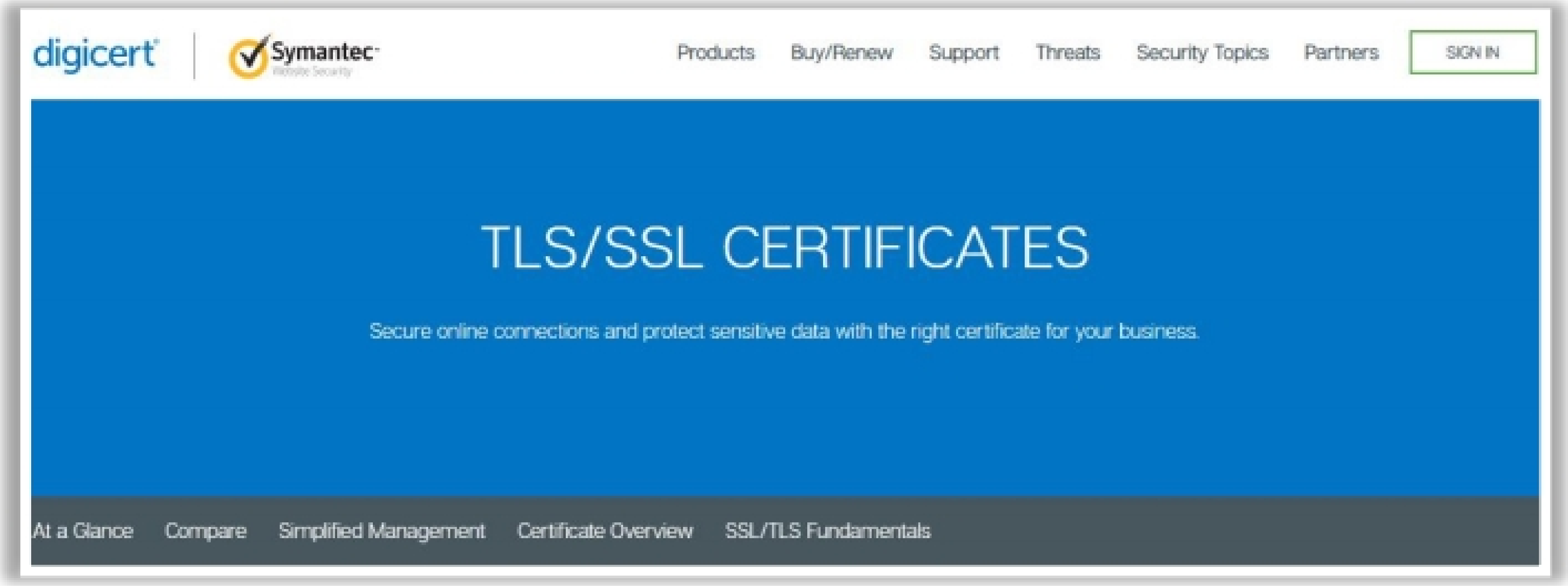

digicert
Symantec
Products
Buy/Renew
Support
Threats
Security Topics
Partners
SIGN IN
TLS/SSL CERTIFICATES
Secure online connections and protect sensitive data with the right certificate for your business.
At a Glance
Compare
Simplified Management
Certificate Overview
SSL/TLS Fundamentals

Figure 20.20: Screenshot of Symantec Website

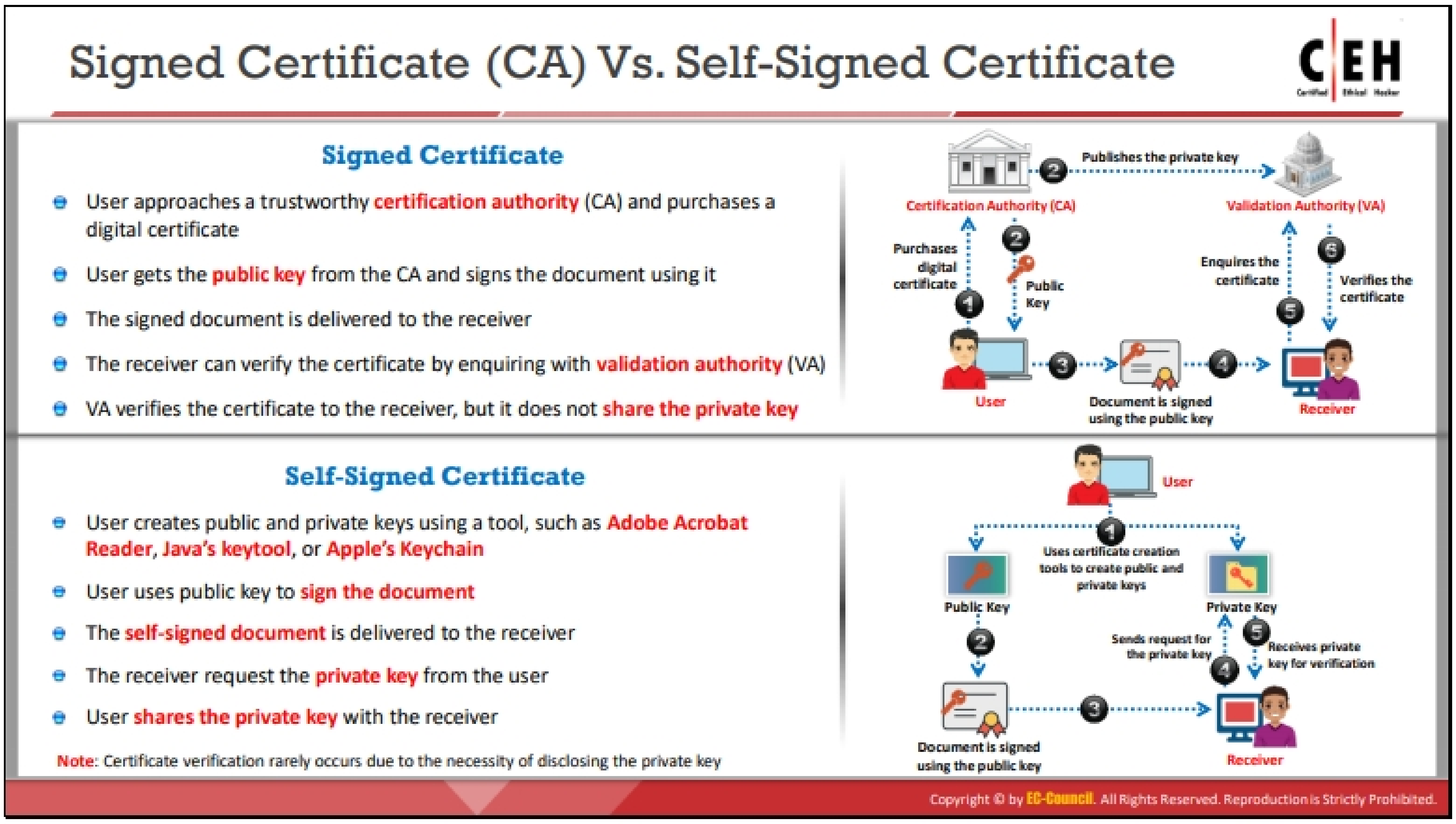
- **GoDaddy**

  Source: *https://www.godaddy.com*

  GoDaddy SSL Certificates offer a complete range of certificates that comply with CA/Browser Forum guidelines. They provide the SHA-2 hash algorithm and 2048-bit encryption, protection of unlimited servers, etc.

GoDaddy
Help by GoDaddy Guides
Sign In
SSL Certificates
Get an SSL certificate.
Show visitors you're trustworthy and authentic.
Call for a free security assessment: (480) 463-8327

Figure 20.21: Screenshot of GoDaddy Website



Signed Certificate (CA) Vs. Self-Signed Certificate

Signed Certificate

- User approaches a trustworthy **certification authority** (CA) and purchases a digital certificate
- User gets the **public key** from the CA and signs the document using it
- The signed document is delivered to the receiver
- The receiver can verify the certificate by enquiring with **validation authority** (VA)
- VA verifies the certificate to the receiver, but it does not **share the private key**

Publishes the private key
2
Certification Authority (CA)
Validation Authority (VA)
Purchases digital certificate
1
2
Public Key
Enquires the certificate
5
6
Verifies the certificate
User
3
Document is signed using the public key
4
Receiver

Self-Signed Certificate

- User creates public and private keys using a tool, such as **Adobe Acrobat Reader**, **Java's keytool**, or **Apple's Keychain**
- User uses public key to **sign the document**
- The **self-signed document** is delivered to the receiver
- The receiver request the **private key** from the user
- User **shares the private key** with the receiver

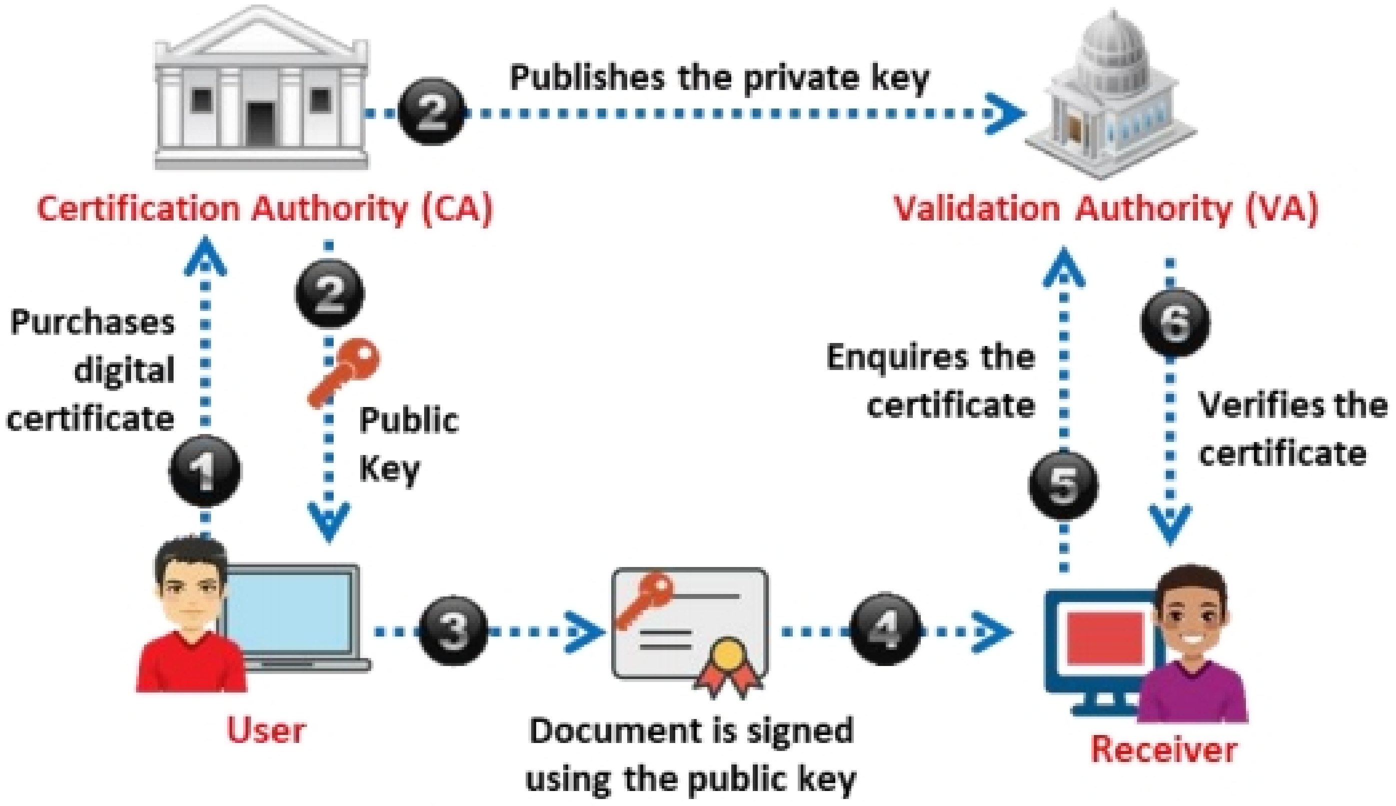
Note: Certificate verification rarely occurs due to the necessity of disclosing the private key

User
1
Uses certificate creation tools to create public and private keys
Public Key
Private Key
2
Sends request for the private key
4
5
Receives private key for verification
3
Document is signed using the public key
Receiver

Signed Certificate (CA) vs. Self-Signed Certificate

- **Signed Certificate**

  CAs sign and issue signed certificates. These certificates contain a public key and the identity of the owner. The corresponding private key is not made publicly available; instead, it is kept secret by the authorized user. By issuing the certificate, the CA confirms or validates that the public key contained in the certificate belongs to the person, company, server, or other entity mentioned in the certificate. CA verifies an application's credentials; thus, users and relying parties trust the information in the CA's certificates. The CA accepts responsibility for saying, "Yes, this person is who they state they are, and we, the CA, certify that." Some popular CAs include Comodo, IdenTrust, Symantec, and GoDaddy.



2
Publishes the private key
Certification Authority (CA)
Validation Authority (VA)
Purchases digital certificate
1
2
Public Key
Enquires the certificate
5
6
Verifies the certificate
User
3
Document is signed using the public key
4
Receiver

Figure 20.22: Process of obtaining signed certificates

As shown in the diagram above, the user approaches a trustworthy CA and purchases a digital certificate. The CA issues a certificate (having a public key and the identity of the user) to the user and updates the VA with the same information. The user signs a document with the public key and sends it to the receiver. Then, the receiver verifies the certificate by inquiring with the VA, who verifies the certificate to the receiver but does not share the private key.

- **Self-Signed Certificate**

  A self-signed certificate is an identity certificate signed by the same entity whose identity it certifies. In general, self-signed certificates are widely used for testing servers. In a self-signed certificate, a user creates a pair of public and private keys using a certificate creation tool such as Adobe Acrobat Reader, Java's keytool, and Apple's Keychain, and signs the document with the public key. The receiver requests the sender for the private key to verify the certificate. However, certificate verification rarely occurs due to the necessity of disclosing the private key. This makes self-signed certificates useful only in a self-controlled testing environment.

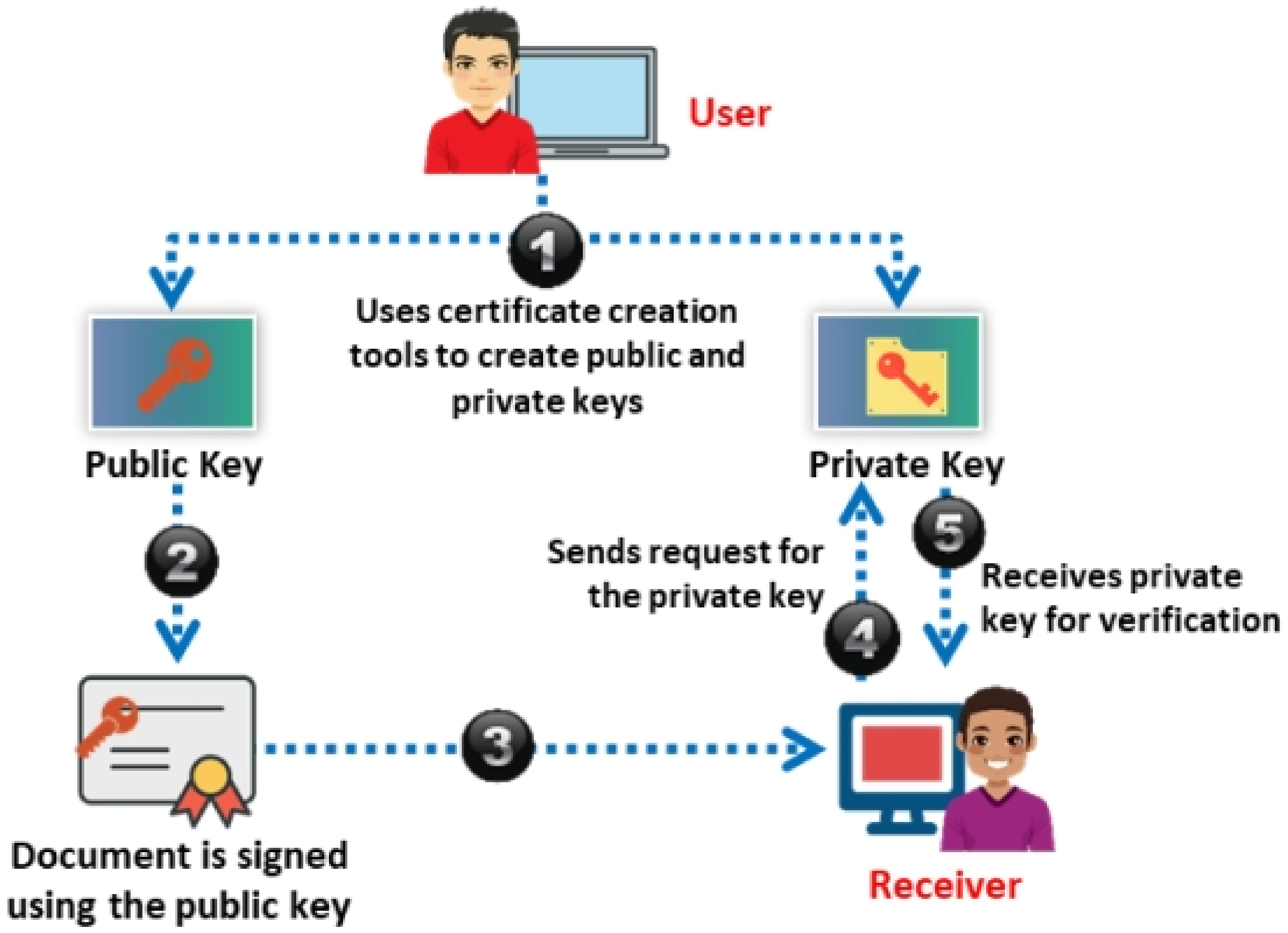
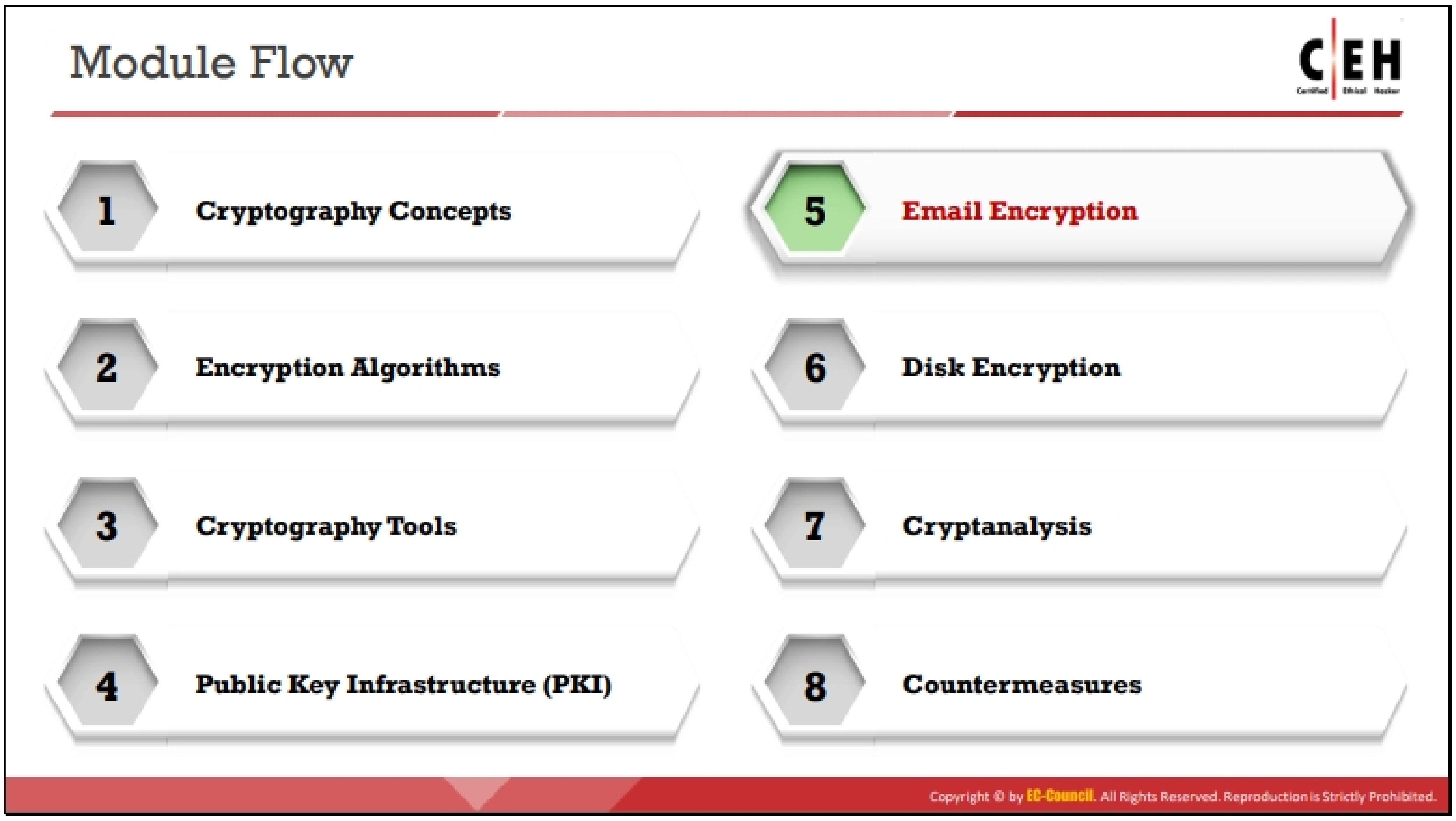

User
1
Uses certificate creation tools to create public and private keys
Public Key
Private Key
2
Sends request for the private key
4
5
Receives private key for verification
3
Document is signed using the public key
Receiver

Figure 20.23: Process of generating self-signed certificates



Module Flow

1 **Cryptography Concepts**

2 **Encryption Algorithms**

3 **Cryptography Tools**

4 **Public Key Infrastructure (PKI)**

5 **Email Encryption**

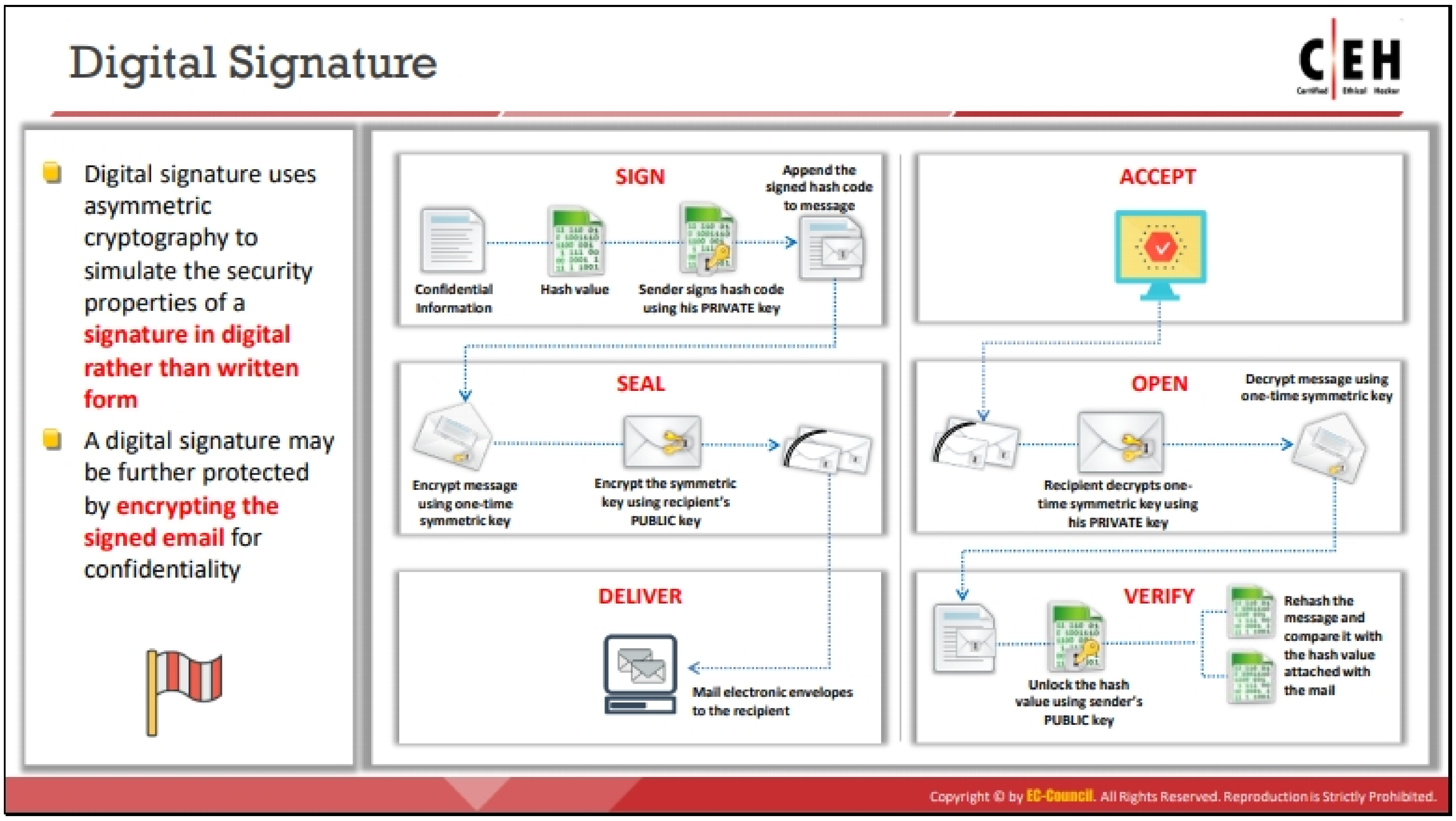
6 **Disk Encryption**

7 **Cryptanalysis**

8 **Countermeasures**

Email Encryption

Currently, most businesses use email as the primary source of communication, as it is simple and easy to communicate or share information. Emails can contain sensitive information about the organization, such as projects, upcoming news, and financial data, which, when accessed by the wrong person, can cause severe losses to the organization. One can protect emails containing sensitive information by encrypting them. This section deals with email security mechanisms such as digital signature, SSL, TLS, cryptographic toolkits, PGP, GPG, and email encryption tools.



Digital Signature

- Digital signature uses asymmetric cryptography to simulate the security properties of a **signature in digital rather than written form**
- A digital signature may be further protected by **encrypting the signed email** for confidentiality

SIGN
Confidential Information
Hash value
Sender signs hash code using his PRIVATE key
Append the signed hash code to message
SEAL
Encrypt message using one-time symmetric key
Encrypt the symmetric key using recipient's PUBLIC key
DELIVER
Mail electronic envelopes to the recipient
ACCEPT
OPEN
Decrypt message using one-time symmetric key
Recipient decrypts one-time symmetric key using his PRIVATE key
VERIFY
Unlock the hash value using sender's PUBLIC key
Rehash the message and compare it with the hash value attached with the mail

Digital Signature

A digital signature uses asymmetric cryptography to simulate the security properties of a signature in digital form rather than in written form. A digital signature is a cryptographic means of authentication. Public-key cryptography uses asymmetric encryption and helps the user to create a digital signature. The two types of keys in public-key cryptography are the private key (only the signer knows this key and uses it to create a digital signature) and the public key (it is widely known and the relying party uses it to verify the digital signature).

A hash function is an algorithm that helps a user to create and verify a digital signature. This algorithm creates a digital representation, also known as the message fingerprint. This fingerprint has a hash value that is much smaller than the message, but one that is unique to it. If the attacker changes the message, the hash function will automatically produce a different hash value.

To verify the digital signature, one needs the hash value of the original message and the encryption algorithm used to create the digital signature. Using both the public key and the new result, the verifier checks to see if the digital signature was created with the related private key and whether the new hash value is the same as the original one. A digital signature may be further protected by encrypting the signed email for confidentiality.

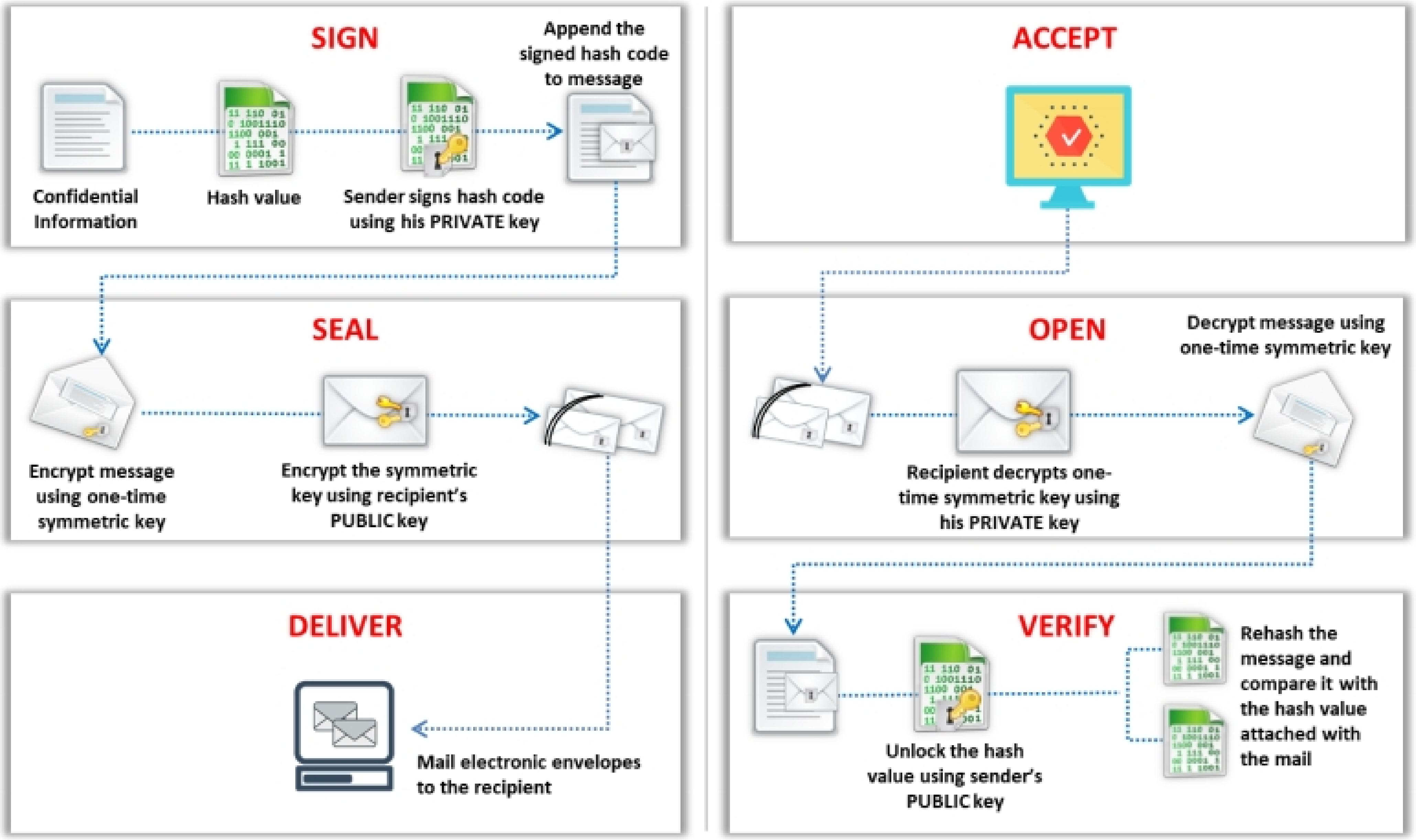
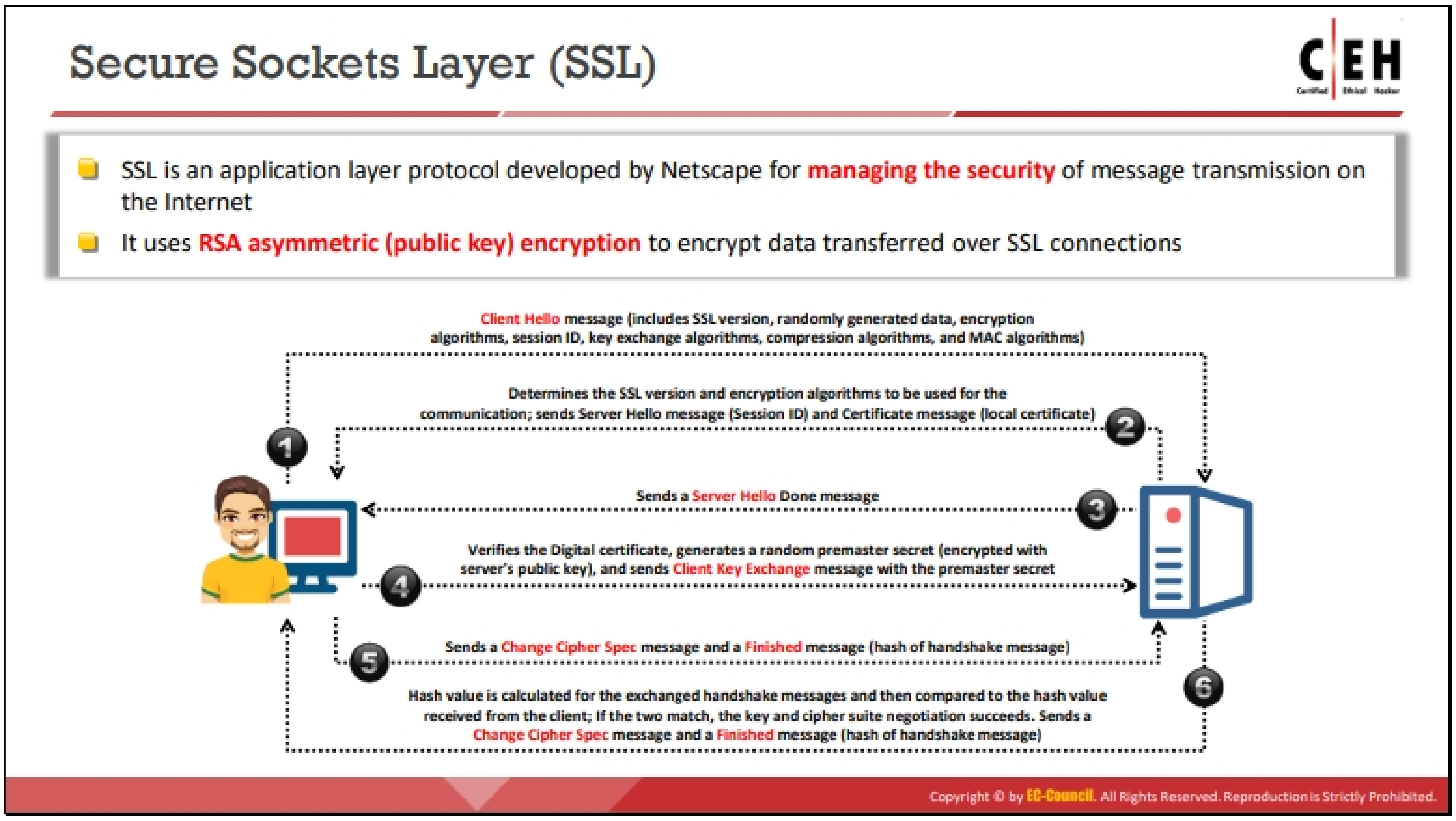

SIGN
Append the signed hash code to message
Confidential Information
Hash value
Sender signs hash code using his PRIVATE key
ACCEPT
SEAL
Encrypt message using one-time symmetric key
Encrypt the symmetric key using recipient's PUBLIC key
OPEN
Decrypt message using one-time symmetric key
Recipient decrypts one-time symmetric key using his PRIVATE key
DELIVER
Mail electronic envelopes to the recipient
VERIFY
Unlock the hash value using sender's PUBLIC key
Rehash the message and compare it with the hash value attached with the mail

Figure 20.24: Using digital signature for email security



Secure Sockets Layer (SSL)

- SSL is an application layer protocol developed by Netscape for **managing the security** of message transmission on the Internet
- It uses **RSA asymmetric (public key) encryption** to encrypt data transferred over SSL connections

1. Client Hello message (includes SSL version, randomly generated data, encryption algorithms, session ID, key exchange algorithms, compression algorithms, and MAC algorithms)
2. Determines the SSL version and encryption algorithms to be used for the communication; sends Server Hello message (Session ID) and Certificate message (local certificate)
3. Sends a Server Hello Done message
4. Verifies the Digital certificate, generates a random premaster secret (encrypted with server's public key), and sends Client Key Exchange message with the premaster secret
5. Sends a Change Cipher Spec message and a Finished message (hash of handshake message)
6. Hash value is calculated for the exchanged handshake messages and then compared to the hash value received from the client; If the two match, the key and cipher suite negotiation succeeds. Sends a Change Cipher Spec message and a Finished message (hash of handshake message)

Secure Sockets Layer (SSL)

The Secure Sockets Layer (SSL) protocol is an application layer protocol developed by Netscape for managing the security of message transmission on the Internet. It is used to provide a secure authentication mechanism between two communicating applications, such as a client and a server. SSL requires a reliable transport protocol, such as TCP, for data transmission and reception. It uses RSA asymmetric (public-key) encryption to encrypt data transferred over SSL connections.

Any application-layer protocol that is higher than SSL, such as HTTP, FTP, and telnet, can form a transparent layer over SSL. SSL acts as an arbitrator between the encryption algorithm and the session key; it also verifies the destination server prior to the transmission and reception of data. SSL encrypts the complete data of the application protocol to ensure security.

SSL also offers "**channelsecurity**" with three basic properties:

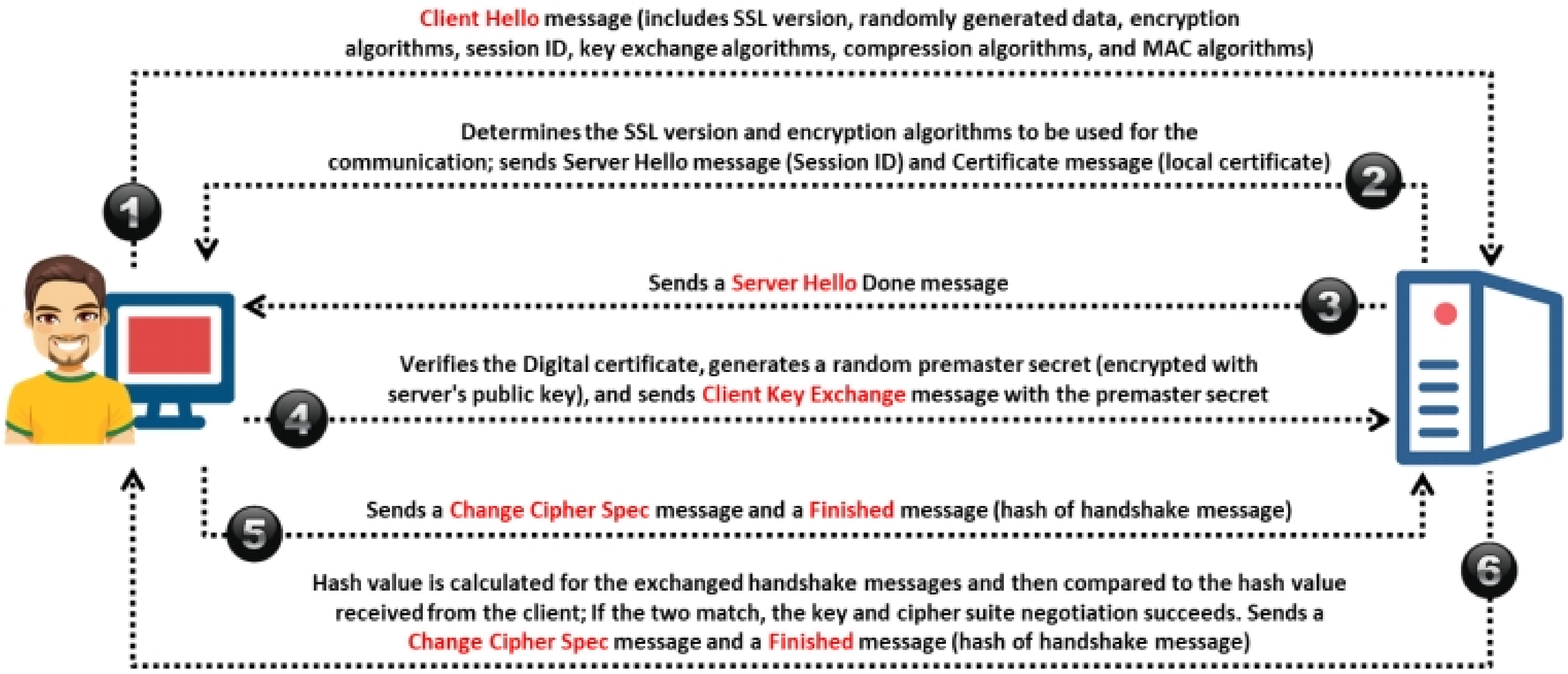
- **Private channel** – All the messages are encrypted after a simple handshake is used to define a secret key.
- **Authenticated channel** – The server endpoint of the conversation is always encrypted, whereas the client endpoint is optionally authenticated.
- **Reliable channel** – Message transfer has an integrity check.

SSL uses both asymmetric and symmetric authentication mechanisms. Public-key encryption verifies the identities of the server, the client, or both. Once authentication has occurred, the client and server can create symmetric keys, allowing them to communicate and transfer data rapidly. An SSL session is responsible for carrying out the SSL handshake protocol to organize the states of the server and clients, thus ensuring consistency of the protocol.

SSL Handshake Protocol Flow

The SSL handshake protocol works on top of the SSL record layer. The processes executed in the three-way handshake protocol are as follows:

1. The client sends a hello message to the server, to which the server must respond with a hello message, or the connection will fail due to the occurrence of a fatal error. The attributes established due to the server and client hello are protocol version, session ID, cipher suite, and compression method.
2. After the connection is established, the server sends a certificate to the client for authentication. In addition, the server might send a server-key exchange message. On authentication of the server, it may ask the client for the certificate (if appropriate for the cipher suite selected).
3. The server sends a "hello done" message to inform the client that the handshake phase is complete and waits for the client's response.
4. If the client receives a certificate-request message, the client must respond to the message by sending a certificate message or "no certificate" alert. The server sends the client key-exchange message. The content of the message depends on the public-key algorithm between the server hello and the client hello. If the certificate sent by the client has signing ability, a digitally signed certificate verifies the message, and the client transmits it.
5. The client transmits the changed cipher-spec message and copies the pending cipher spec into the current cipher spec. The client sends a message to initiate the completion of the message under the new algorithm, keys, and secrets.
6. In response, the server replies by sending its own changed cipher-spec message, transfers the pending cipher spec to the current cipher spec, and initiates the completion of the message under the new cipher spec. At this point, the handshake is complete and the server starts exchanging the application-layer data.



Client Hello message (includes SSL version, randomly generated data, encryption algorithms, session ID, key exchange algorithms, compression algorithms, and MAC algorithms)

Determines the SSL version and encryption algorithms to be used for the communication; sends Server Hello message (Session ID) and Certificate message (local certificate)

Sends a Server Hello Done message

Verifies the Digital certificate, generates a random premaster secret (encrypted with server's public key), and sends Client Key Exchange message with the premaster secret

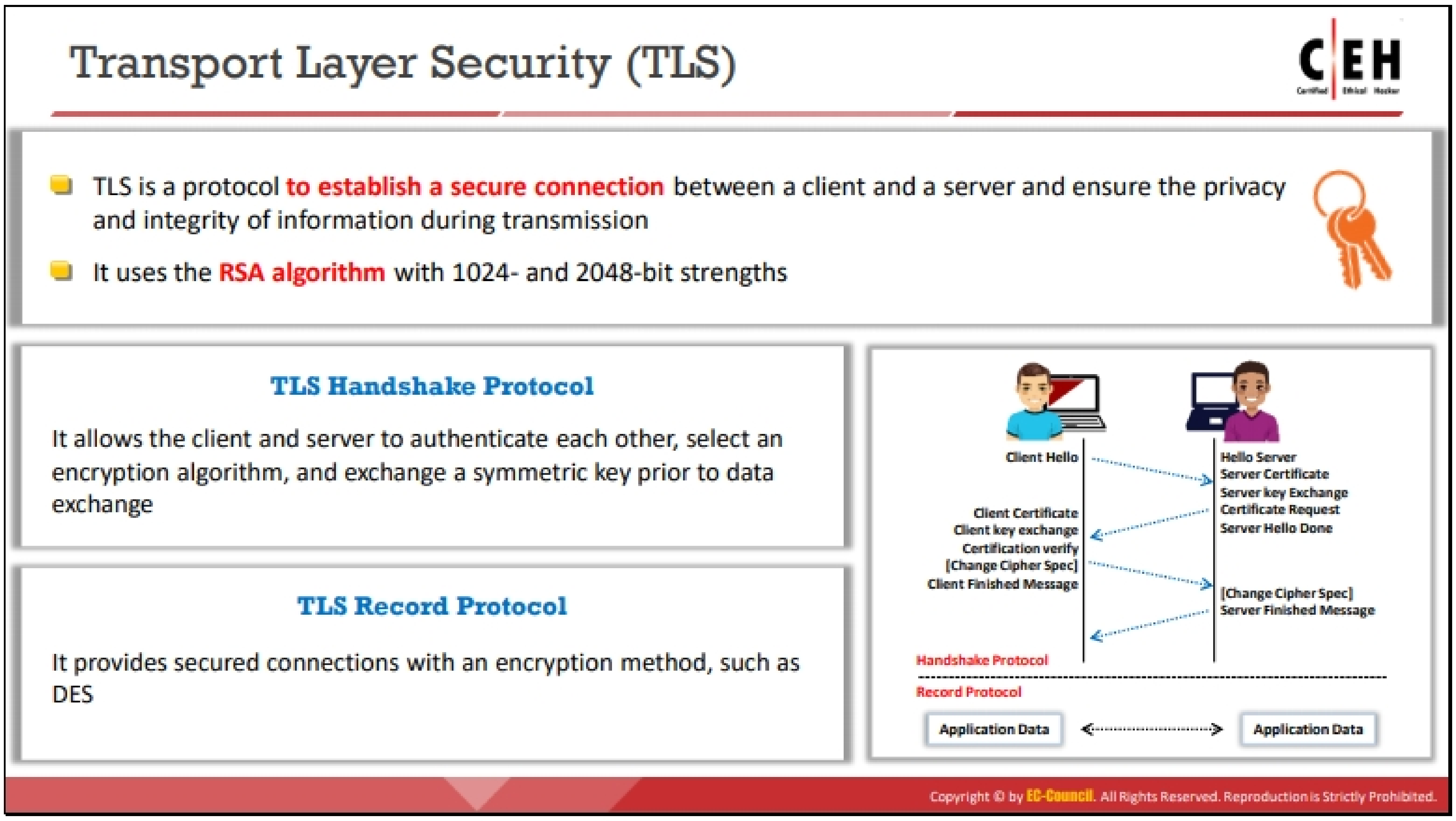
Sends a Change Cipher Spec message and a Finished message (hash of handshake message)

Hash value is calculated for the exchanged handshake messages and then compared to the hash value received from the client; If the two match, the key and cipher suite negotiation succeeds. Sends a Change Cipher Spec message and a Finished message (hash of handshake message)

Figure 20.25: SSL handshake protocol flow

The resumption of a previous session or the replication of an existing session proceeds as follows:

- The client initiates the communication by sending a hello message with the session ID of the session that is to be resumed.
- If the server finds a match, it re-establishes the session under the specified session state with the same session ID.
- At this point, both the server and the client exchange the changed spec messages and proceed directly to the finished messages.
- After re-establishment, the server and client exchange data at the application layer.
- If the session ID does not exist, the server creates a new session ID. The SSL client and server then carry out a complete handshake.



Transport Layer Security (TLS)

- TLS is a protocol **to establish a secure connection** between a client and a server and ensure the privacy and integrity of information during transmission
- It uses the **RSA algorithm** with 1024- and 2048-bit strengths

TLS Handshake Protocol

It allows the client and server to authenticate each other, select an encryption algorithm, and exchange a symmetric key prior to data exchange

TLS Record Protocol

It provides secured connections with an encryption method, such as DES

Client Hello
Hello Server
Server Certificate
Server key Exchange
Certificate Request
Server Hello Done
Client Certificate
Client key exchange
Certification verify
[Change Cipher Spec]
Client Finished Message
[Change Cipher Spec]
Server Finished Message
Handshake Protocol
Record Protocol
Application Data
Application Data

Transport Layer Security (TLS)

The Transport Layer Security (TLS) protocol is used to establish a secure connection between a client and a server and ensure the privacy and integrity of information during transmission. It uses a symmetric key for bulk encryption, asymmetric key for authentication and key exchange, and message authentication codes for message integrity. It uses the RSA algorithm with strengths of 1024 and 2048 bits. Using TLS, one can reduce security risks such as message tampering, message forgery, and message interception. An advantage of TLS is that it is independent of the application protocol. Higher-level protocols can lie on top of TLS transparently.

TLS consists of two layers: TLS Record Protocol and TLS Handshake Protocol.

1. **TLS Record Protocol**

   The TLS Record Protocol is a layered protocol. It provides secured connections with an encryption method such as DES. It secures application data using the keys generated during the handshake and verifies its integrity and origin. The TLS Record Protocol provides connection security with two basic properties:

   - **The connection is private**: Uses symmetric cryptography for data encryption (e.g., DES). The protocol generates unique keys for symmetric encryption for each connection, depending on a secret negotiated by another protocol (such as the TLS Handshake Protocol). One can use the TLS Record Protocol without encryption.
   - **The connection is reliable**: It provides a message integrity check at the time of message transport using a keyed MAC. Secure hash functions (e.g., SHA, MD5) help to perform MAC computations.

The TLS Record Protocol does the following:

- Fragments outgoing data into manageable blocks and reassembles incoming data
- Optionally compresses outgoing data and decompresses incoming data
- Applies MAC to the outgoing data and uses MAC to verify the incoming data
- Encrypts outgoing data and decrypts incoming data

The TLS Record Protocol sends the outgoing encrypted data to the TCP layer for transport.

2. **TLS Handshake Protocol**

The TLS Handshake Protocol allows the client and server to authenticate each other and select an encryption algorithm and cryptographic keys prior to data exchange by the application protocol.

It provides connection security with three basic properties:

- The peer's identity can be authenticated using asymmetric cryptography. This can be made optional but is mostly required for at least one of the peers.
- The negotiation of a shared secret is secure.
- The negotiation is reliable.

The TLS Handshake Protocol operates on top of the TLS Record Protocol and is responsible for producing cryptographic parameters of the session state. At the start of communication, the TLS client and server agree on a protocol version, select cryptographic algorithms, optionally authenticate each other, and use asymmetric cryptography techniques to create shared secrets.

The steps involved in the TLS Handshake Protocol are as follows:

- Initially, the client sends a "Client hello" message accompanied by the client's random value and supported cipher suites to the server.
- The server responds to the client by sending a "Server hello" message accompanied by the server's random value.
- The server sends its certificate to the client for authentication and may request the client's certificate. The server sends the "Server hello done" message.
- The client sends its certificate to the server, if requested.
- The client generates a random pre-master secret and encrypts it with the server's public key; then, it sends the encrypted pre-master secret to the server.
- The server receives the pre-master secret. Thereafter, the client and server each create the master secret and session keys based on the pre-master secret.
- The client sends "Change cipher spec" to the server to indicate that it will start using the new session keys for hashing and encrypting messages. The client also sends "Clientfinished".

- The server receives "Changecipher spec" from the client and switches its record layer security state to symmetric encryption using the session keys. Then, the server sends "Serverfinished" to the client.
- Now, the client and server can exchange application data over the secure channel they have established, and all the messages exchanged between the client and server are encrypted using a session key.

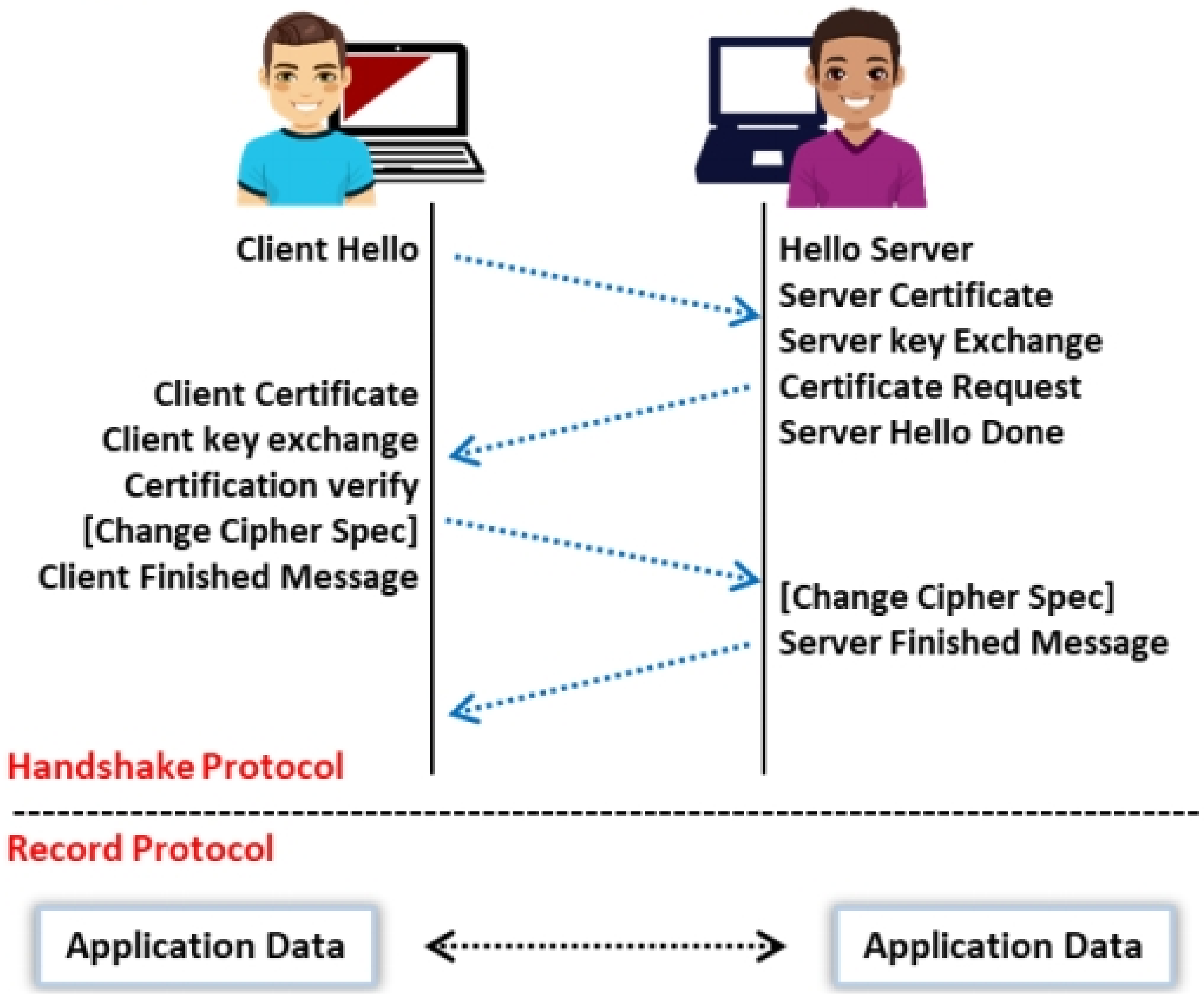
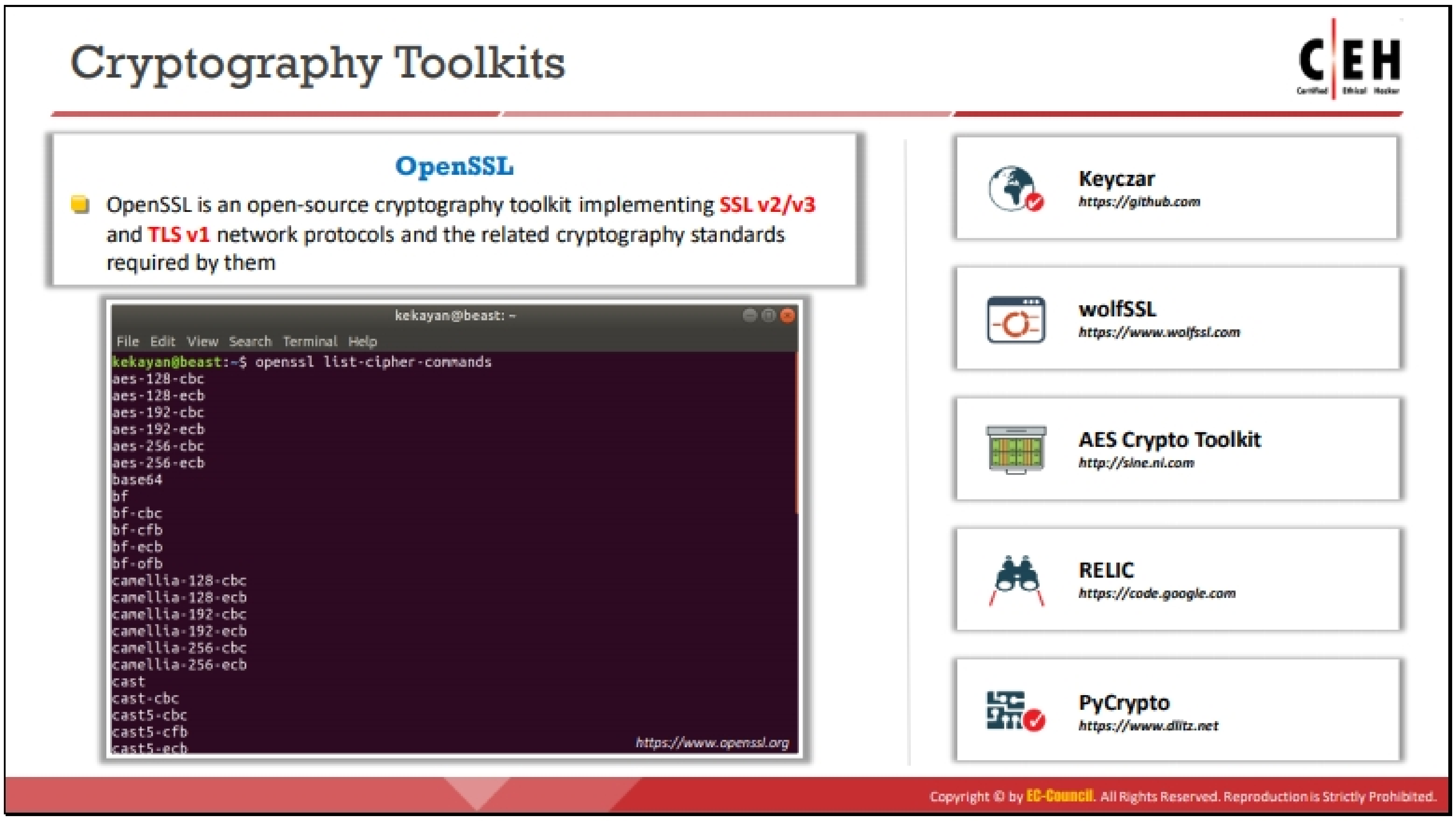

Client Hello
Hello Server
Server Certificate
Server key Exchange
Certificate Request
Server Hello Done
Client Certificate
Client key exchange
Certification verify
[Change Cipher Spec]
Client Finished Message
[Change Cipher Spec]
Server Finished Message
Handshake Protocol
Record Protocol
Application Data
Application Data

Figure 20.26: TLS handshake and record protocols



Cryptography Toolkits

OpenSSL

- OpenSSL is an open-source cryptography toolkit implementing **SSL v2/v3** and **TLS v1** network protocols and the related cryptography standards required by them

```
kekayan@beast:~$ openssl list-cipher-commands
aes-128-cbc
aes-128-ecb
aes-192-cbc
aes-192-ecb
aes-256-cbc
aes-256-ecb
base64
bf
bf-cbc
bf-cfb
bf-ecb
bf-ofb
camellia-128-cbc
camellia-128-ecb
camellia-192-cbc
camellia-192-ecb
camellia-256-cbc
camellia-256-ecb
cast
cast-cbc
cast5-cbc
cast5-cfb
cast5-ecb
```

https://www.openssl.org

Keyczar
https://github.com

wolfSSL
https://www.wolfssl.com

AES Crypto Toolkit
http://sine.ni.com

RELIC
https://code.google.com

PyCrypto
https://www.dlitz.net

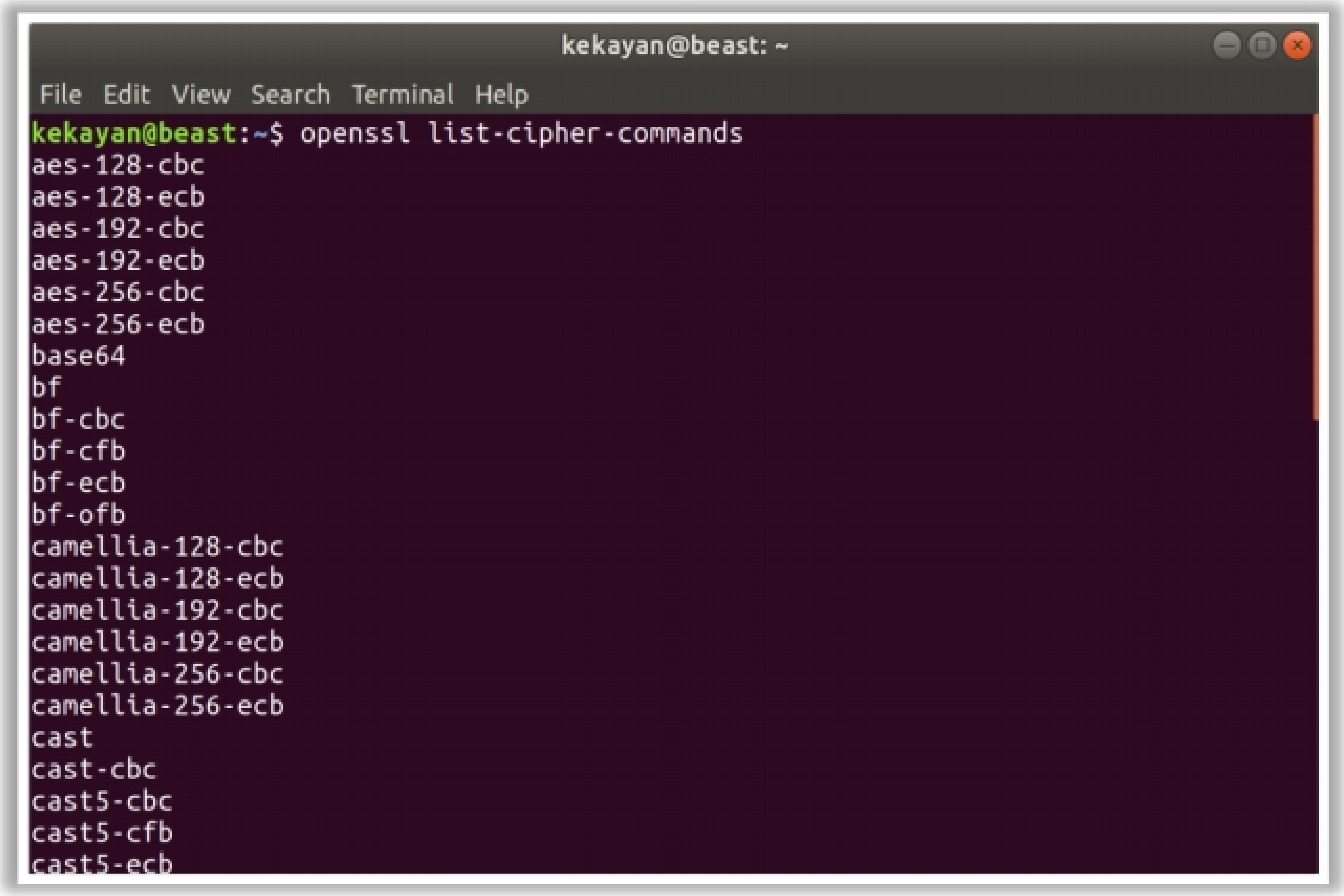
Cryptography Toolkits

Cryptography toolkits include cryptographic primitives, algorithms, and schemes used to provide security for various applications. Some cryptography toolkits are discussed below:

- **OpenSSL**

  Source: *https://www.openssl.org*

  OpenSSL is an open-source cryptography toolkit implementing the SSL and TLS network protocols and the related cryptography standards required by them. It is a command-line tool for using the various cryptography functions of OpenSSL's crypto-library from the shell. OpenSSL can be used for the creation and management of private keys, public keys, and parameters; public-key cryptographic operations; creation of X.509 certificates, CSRs, and CRLs; etc.

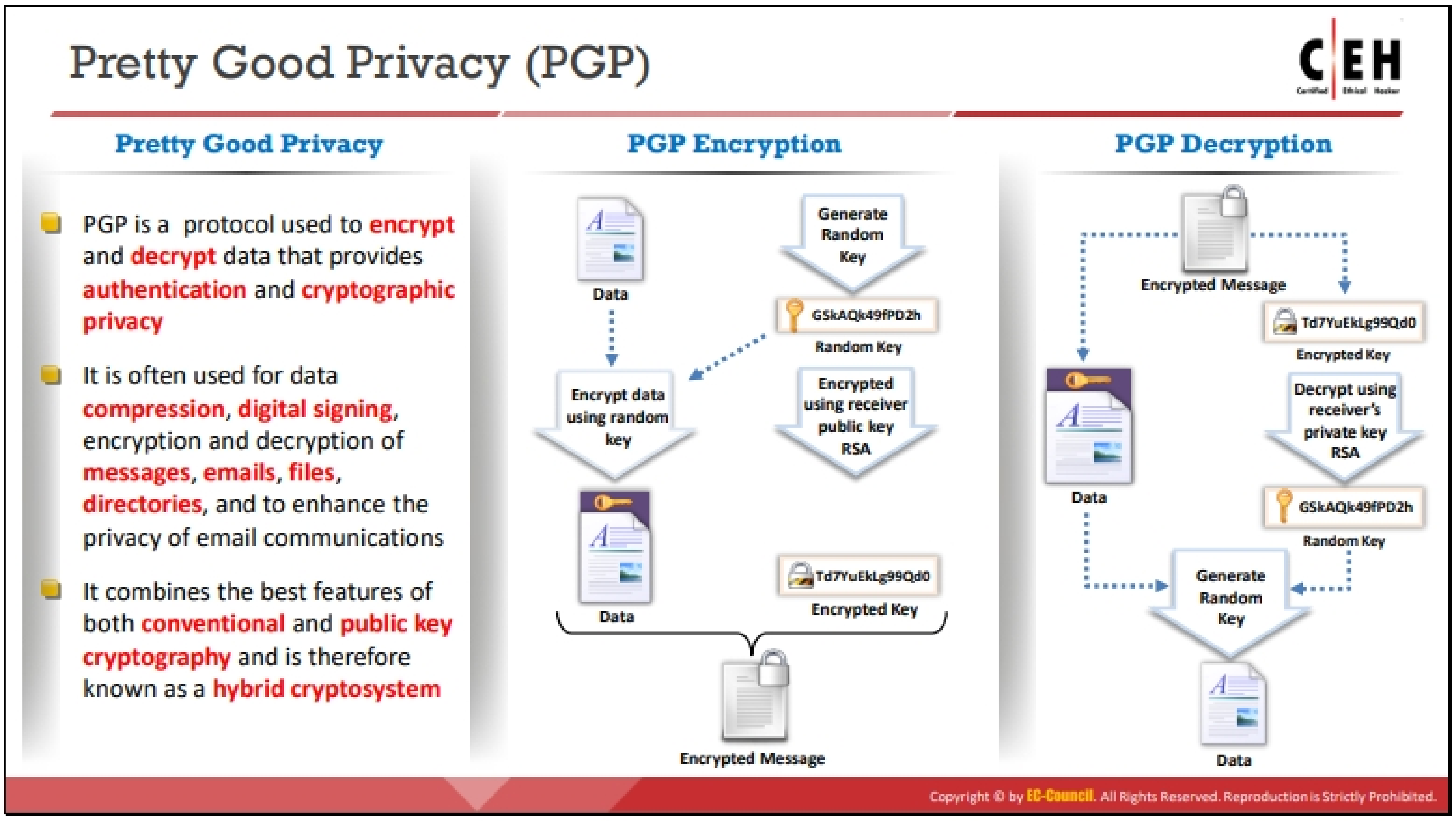


```
kekayan@beast:~$ openssl list-cipher-commands
aes-128-cbc
aes-128-ecb
aes-192-cbc
aes-192-ecb
aes-256-cbc
aes-256-ecb
base64
bf
bf-cbc
bf-cfb
bf-ecb
bf-ofb
camellia-128-cbc
camellia-128-ecb
camellia-192-cbc
camellia-192-ecb
camellia-256-cbc
camellia-256-ecb
cast
cast-cbc
cast5-cbc
cast5-cfb
cast5-ecb
```

Figure 20.27: Screenshot of the OpenSSL command-line tool

Some additional cryptography toolkits are as follows:

- Keyczar (*https://github.com*)
- wolfSSL (*https://www.wolfssl.com*)
- AES Crypto Toolkit (*http://sine.ni.com*)
- RELIC (*https://code.google.com*)
- PyCrypto (*https://www.dlitz.net*)



Pretty Good Privacy (PGP)

Pretty Good Privacy

- PGP is a protocol used to **encrypt** and **decrypt** data that provides **authentication** and **cryptographic privacy**
- It is often used for data **compression**, **digital signing**, encryption and decryption of **messages**, **emails**, **files**, **directories**, and to enhance the privacy of email communications
- It combines the best features of both **conventional** and **public key cryptography** and is therefore known as a **hybrid cryptosystem**

PGP Encryption

Data
Generate Random Key
GSkAQk49fPD2h
Random Key
Encrypt data using random key
Encrypted using receiver public key RSA
Data
Td7YuEkLg99Qd0
Encrypted Key
Encrypted Message

PGP Decryption

Encrypted Message
Td7YuEkLg99Qd0
Encrypted Key
Data
Decrypt using receiver's private key RSA
GSkAQk49fPD2h
Random Key
Generate Random Key
Data

Pretty Good Privacy (PGP)

Pretty Good Privacy (PGP) is a protocol used to encrypt and decrypt data with authentication and cryptographic privacy. It is often used for data compression, digital signing, encryption and decryption of messages, emails, files, and directories, and to enhance the privacy of email communications. The algorithm used for message encryption is RSA for key transport and IDEA for bulk-message encryption. PGP uses RSA for computing digital signatures and MD5 for computing message digests.

It combines the best features of both conventional (around 1,000 times faster than public-key encryption) and public-key cryptography (solution to key distribution and data transmission issues), and is therefore known as a hybrid cryptosystem.

PGP is used for:

- Encrypting a message or file prior to transmission so that only the recipient can decrypt and read it
- Clear signing of the plaintext message to ensure the authenticity of the sender
- Encrypting stored computer files so that no one besides the person who encrypted them can decrypt them
- Deleting files rather than just removing them from the directory or folder
- Data compression for storage or transmission

How PGP Works?

- **PGP Encryption**
  - When a user encrypts data with PGP, PGP first compresses the data.

Compressing the data reduces patterns in the plaintext that could be exploited by most cryptanalysis techniques to crack the cipher, thereby increasing the resistance to cryptanalysis considerably.

- PGP then creates a random key (GSkAQk49fPD2h) that is a one-time-only secret key.
- PGP uses the random key generated to encrypt the plaintext, resulting in a ciphertext.
- Once the data are encrypted, a random key is encrypted with the recipient's public key.
- The public-key-encrypted random key (Td7YuEkLg99Qd0) is sent along with the ciphertext to the recipient.

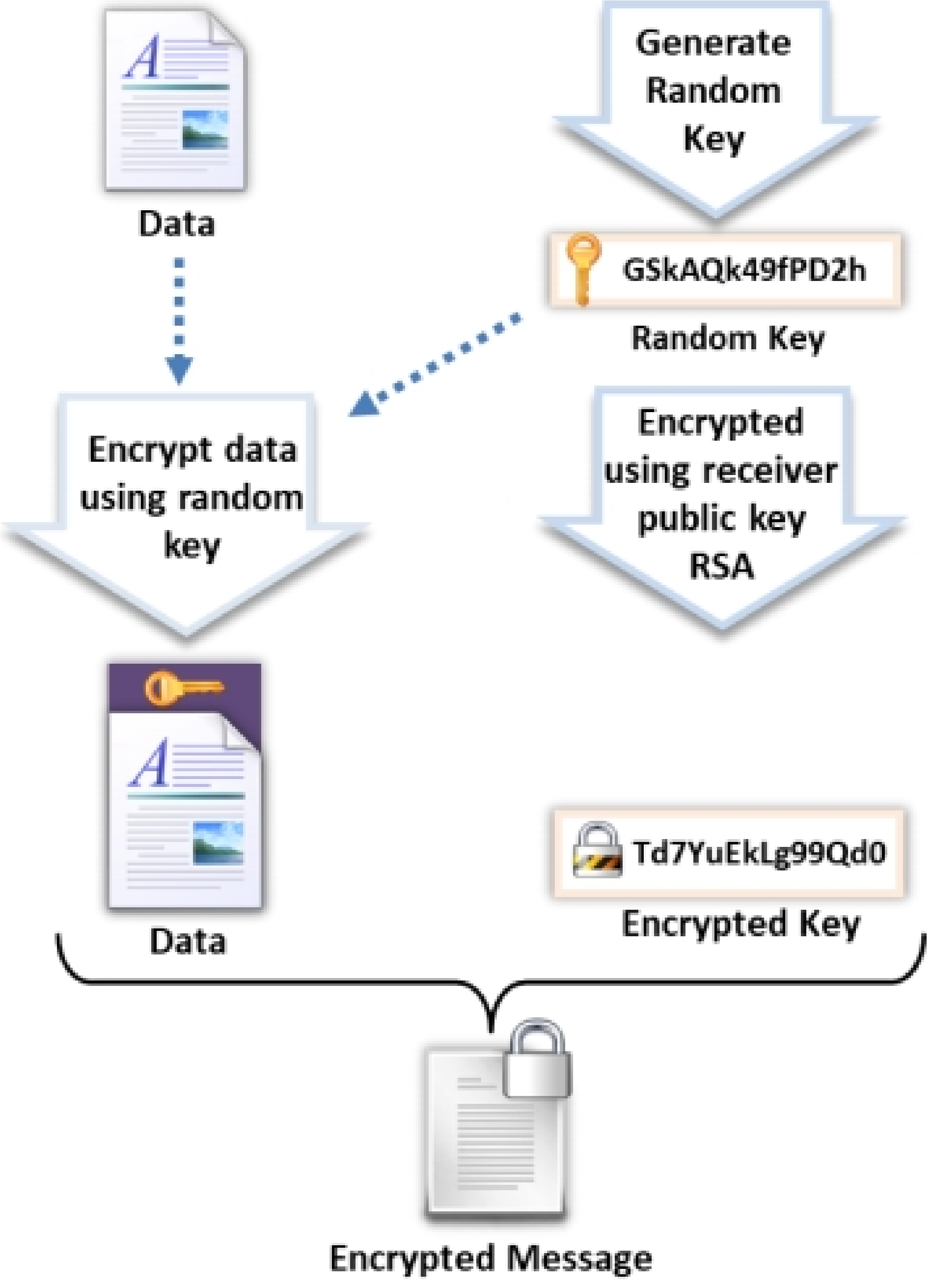

Data
Generate Random Key
GSkAQk49fPD2h
Random Key
Encrypt data using random key
Encrypted using receiver public key RSA
Data
Td7YuEkLg99Qd0
Encrypted Key
Encrypted Message

Figure 20.28: PGP Encryption

- **PGP Decryption**
  - Decryption works in reverse.
  - The recipient's copy of PGP uses his or her private key instead of the public key to recover the temporary random key.
  - PGP then uses the recovered random key to decrypt the conventionally encrypted ciphertext.

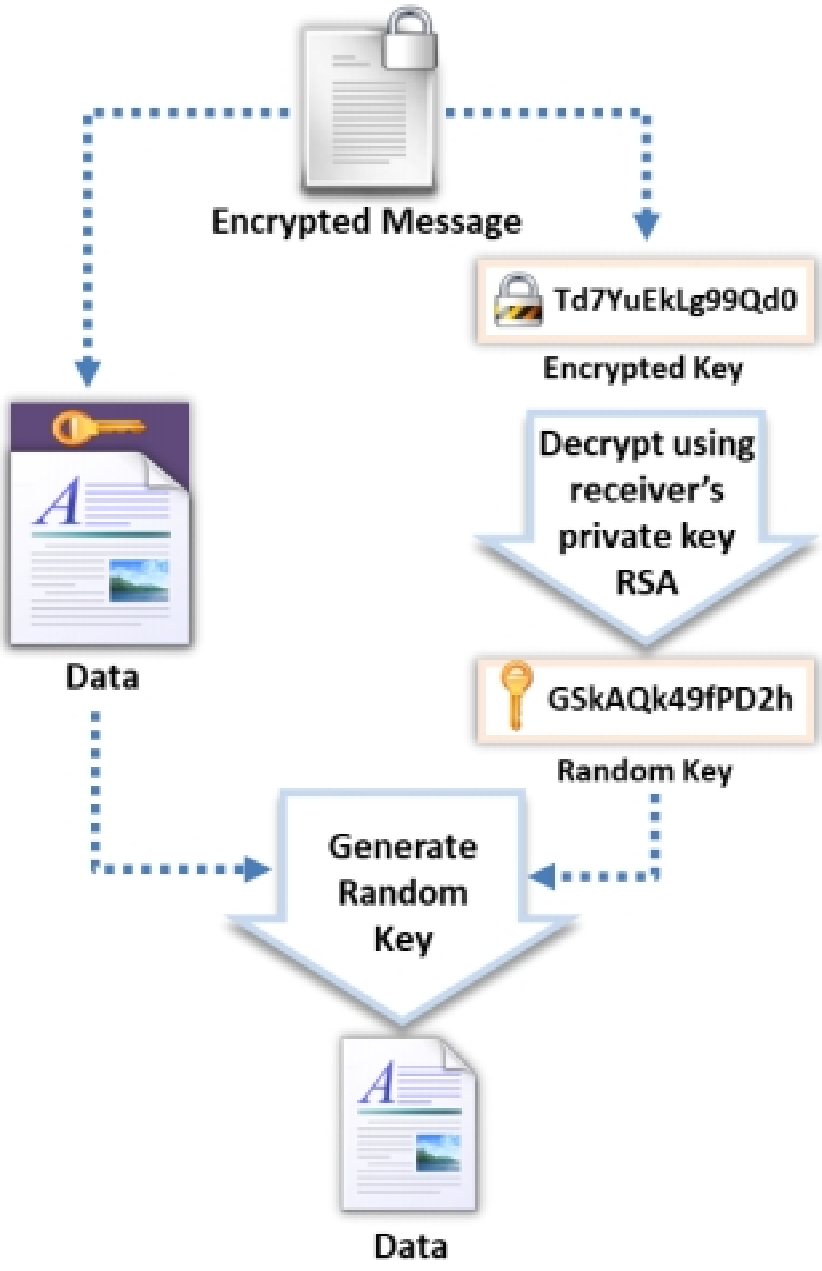
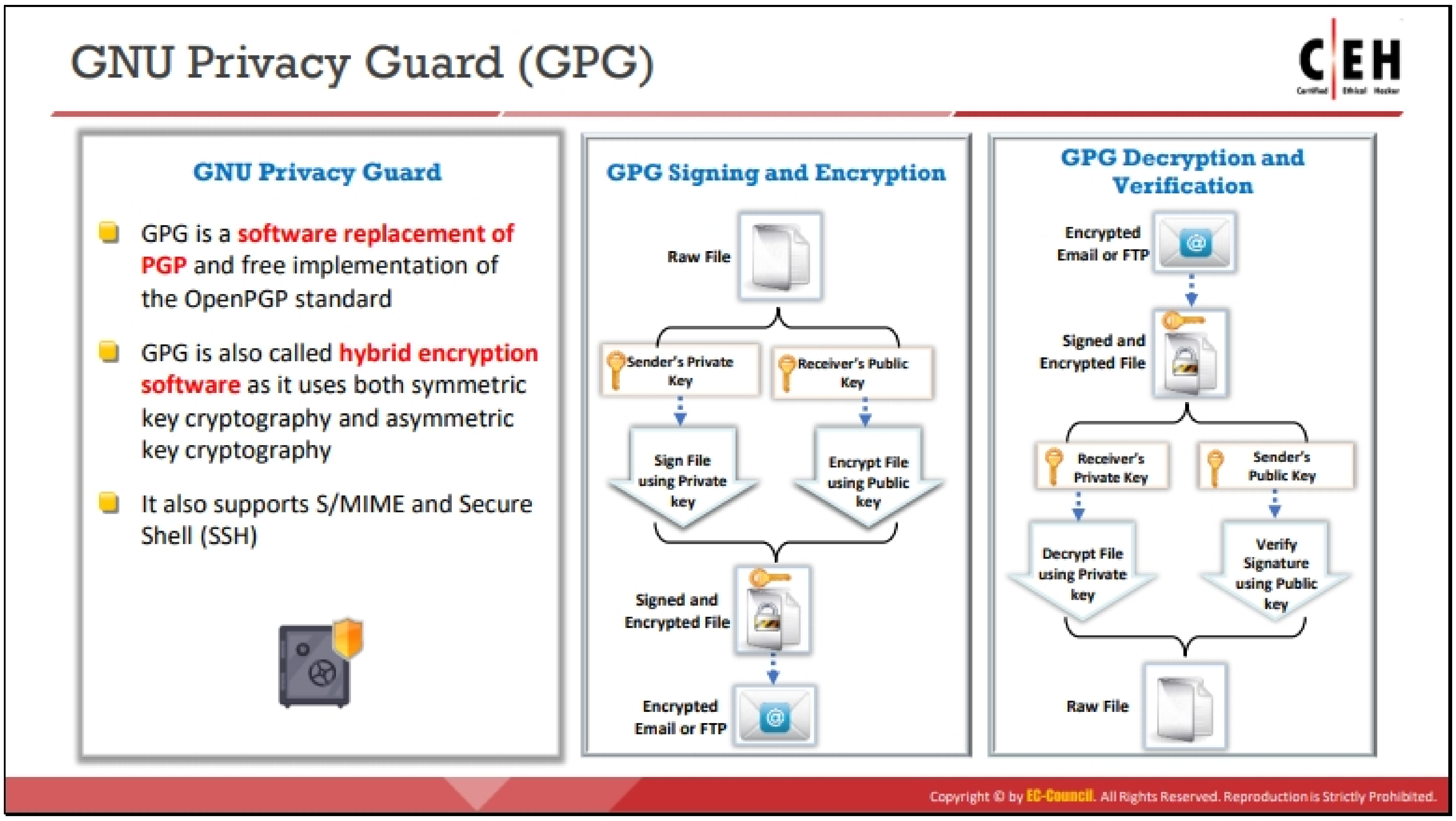

Encrypted Message
Td7YuEkLg99Qd0
Encrypted Key
Decrypt using receiver's private key RSA
Data
GSkAQk49fPD2h
Random Key
Generate Random Key
Data

Figure 20.29: PGP Decryption

Note: Each step of the PGP encryption process (hashing, data compression, symmetric-key cryptography, and public-key cryptography) uses one of the various supported algorithms.



GNU Privacy Guard (GPG)

GNU Privacy Guard

- GPG is a **software replacement of PGP** and free implementation of the OpenPGP standard
- GPG is also called **hybrid encryption software** as it uses both symmetric key cryptography and asymmetric key cryptography
- It also supports S/MIME and Secure Shell (SSH)

GPG Signing and Encryption

Raw File
Sender's Private Key
Receiver's Public Key
Sign File using Private key
Encrypt File using Public key
Signed and Encrypted File
Encrypted Email or FTP

GPG Decryption and Verification

Encrypted Email or FTP
Signed and Encrypted File
Receiver's Private Key
Sender's Public Key
Decrypt File using Private key
Verify Signature using Public key
Raw File

GNU Privacy Guard (GPG)

GNU Privacy Guard (GPG) is a software replacement of PGP and free implementation of the OpenPGP standard that is used to encrypt and decrypt data. GPG is also called a hybrid encryption software program, as it uses both symmetric-key cryptography and asymmetric-key cryptography for improved speed and secure key exchange, which is achieved using the receiver's public key for encrypting the session key.

GPG also supports S/MIME and Secure Shell (SSH). The latest version of GPG supports most cryptographic functions such as elliptic curve cryptography (ECDSA, ECDH, and EdDSA), and it also supports the cryptography library Libgcrypt.

GPG is used for the following:

- Proper key management of both private and public keys
- Creating new private keys and exporting or importing any key even though it is in some armored (e.g., ASCII) format
- Pushing the public key to the key server by signing code with the GPG key having a public signature
- Deleting a private key from local storage
- Encrypting and signing files using asymmetric keys for encrypting any file used for email or FTP
- Decrypting and verifying the encrypted file using asymmetric keys
- Detaching signatures where the signature file can be detached from the message file
- Managing and building the web of trust

- Automatically securing messages in messaging applications such as Psi and Fire

How GPG Works

- **GPG Encryption**
    - GPG encrypts messages individually by using asymmetric-key pairs.
    - The user sends the raw file, and GPG is used for signing the file using the sender's private key for confirming the file content at the time of signing.
    - Then, the file is encrypted using the receiver's public key. Now, the file can be decrypted only with the receiver's private key.
    - After encrypting the data, the encrypted file can be stored locally, distributed to the FTP servers, or sent to email recipients.

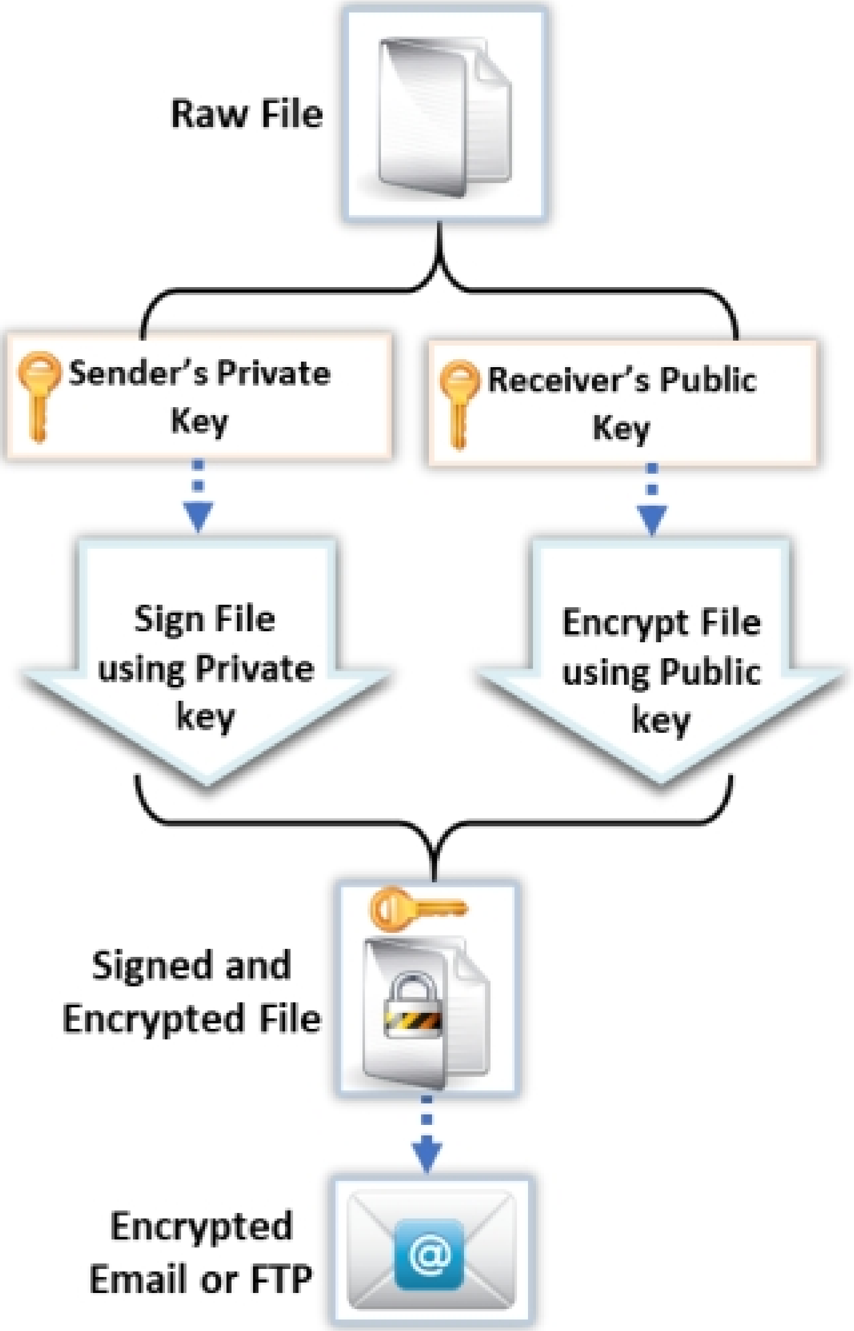

Raw File
Sender's Private Key
Receiver's Public Key
Sign File using Private key
Encrypt File using Public key
Signed and Encrypted File
Encrypted Email or FTP

Figure 20.30: GPG Signing and Encryption

- **GPG Decryption**
    - GPG decryption is the reverse process of GPG encryption.
    - As the asymmetric-key pairs are used, GPG searches for the receiver's private key for decrypting the file.
    - Signature verification is done automatically by the GPG using the sender's public key after the decryption.

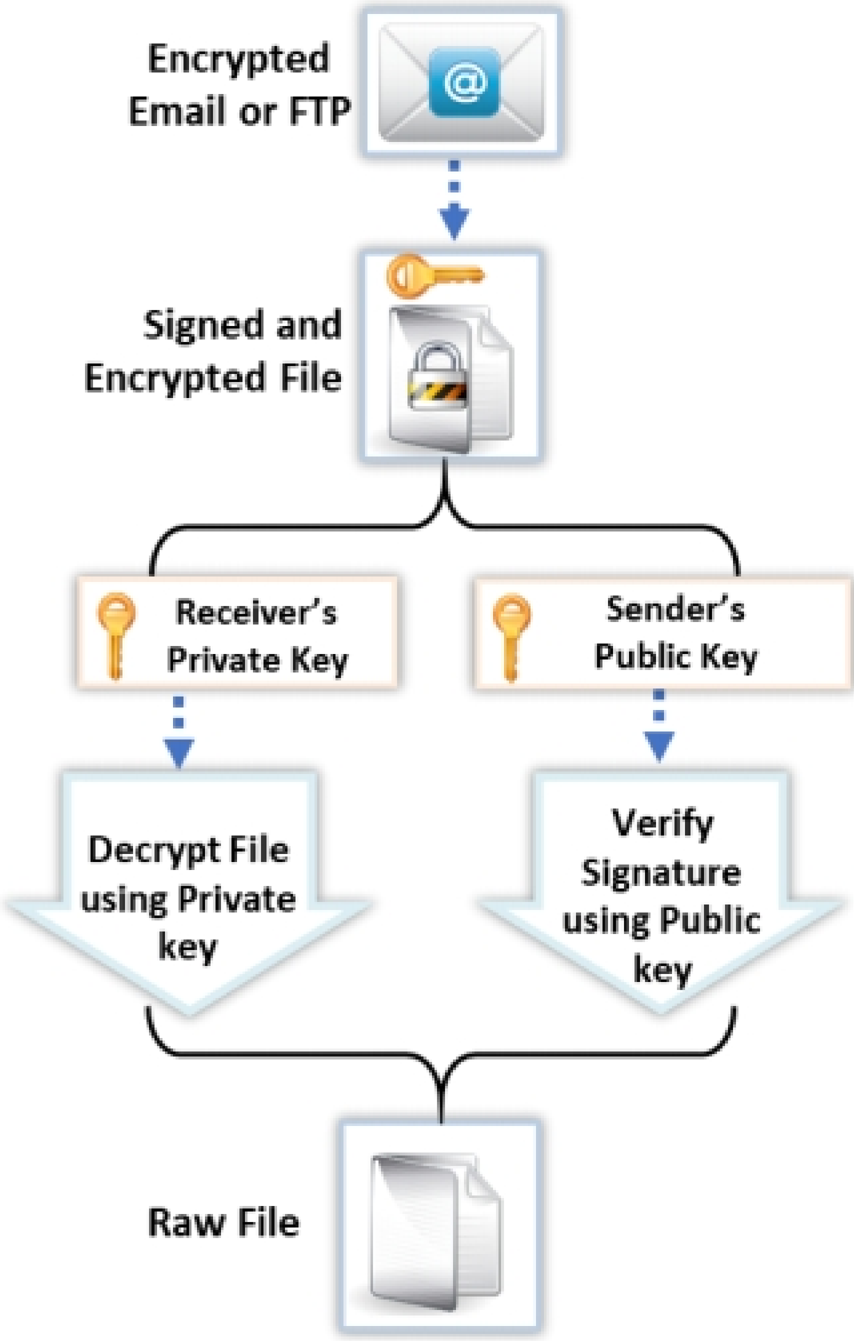
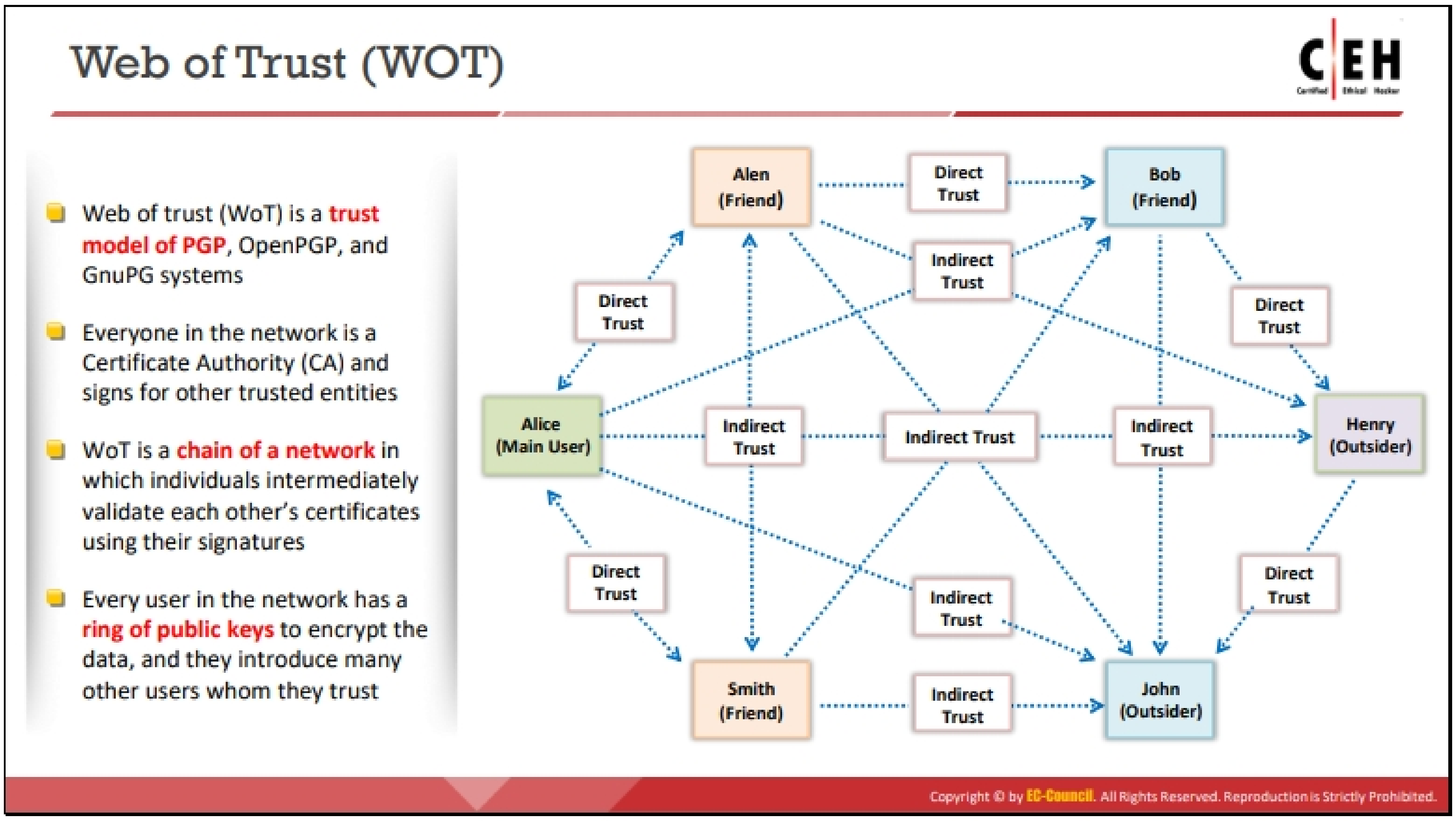

Encrypted Email or FTP
Signed and Encrypted File
Receiver's Private Key
Sender's Public Key
Decrypt File using Private key
Verify Signature using Public key
Raw File

Figure 20.31: GPG Decryption and Verification



Web of Trust (WOT)

- Web of trust (WoT) is a **trust model of PGP**, OpenPGP, and GnuPG systems
- Everyone in the network is a Certificate Authority (CA) and signs for other trusted entities
- WoT is a **chain of a network** in which individuals intermediately validate each other's certificates using their signatures
- Every user in the network has a **ring of public keys** to encrypt the data, and they introduce many other users whom they trust

Alen (Friend)
Direct Trust
Bob (Friend)
Indirect Trust
Direct Trust
Direct Trust
Alice (Main User)
Indirect Trust
Indirect Trust
Indirect Trust
Henry (Outsider)
Direct Trust
Indirect Trust
Direct Trust
Smith (Friend)
Indirect Trust
John (Outsider)

Web of Trust (WOT)

Web of trust (WoT) is a trust model of PGP, OpenPGP, and GnuPG accessible systems. It is an idea of decentralizing the key distribution among PGP users. In the PKI, only a centralized power such as the CA signs certificates in the network, ensuring authenticity between the public key and its owner. In WoT, everyone in the network is a CA, and they can sign for other trusted entities. WoT is a network chain in which individuals intermediately validate each other's certificates using their signatures. These signatures verify the ownership of keys from various trust levels. There is a bunch of similar trust levels through direct or indirect references in WoT.

Working of WOT

In WOT, every PGP user in the network has a ring of public keys to encrypt the data, and they introduce many other users whom they trust. In this trust model, a user encodes the data with the receiver's public key that is decrypted only by the receiver's private key. Then, every user in this model digitally signs the data with their private keys; when the recipient is validating it against the user's public key, he/she can confirm the user's authenticity. This process will ensure that data are received from a valid user without being modified, and only the intended user can access the information, as only he/she holds the related private key.

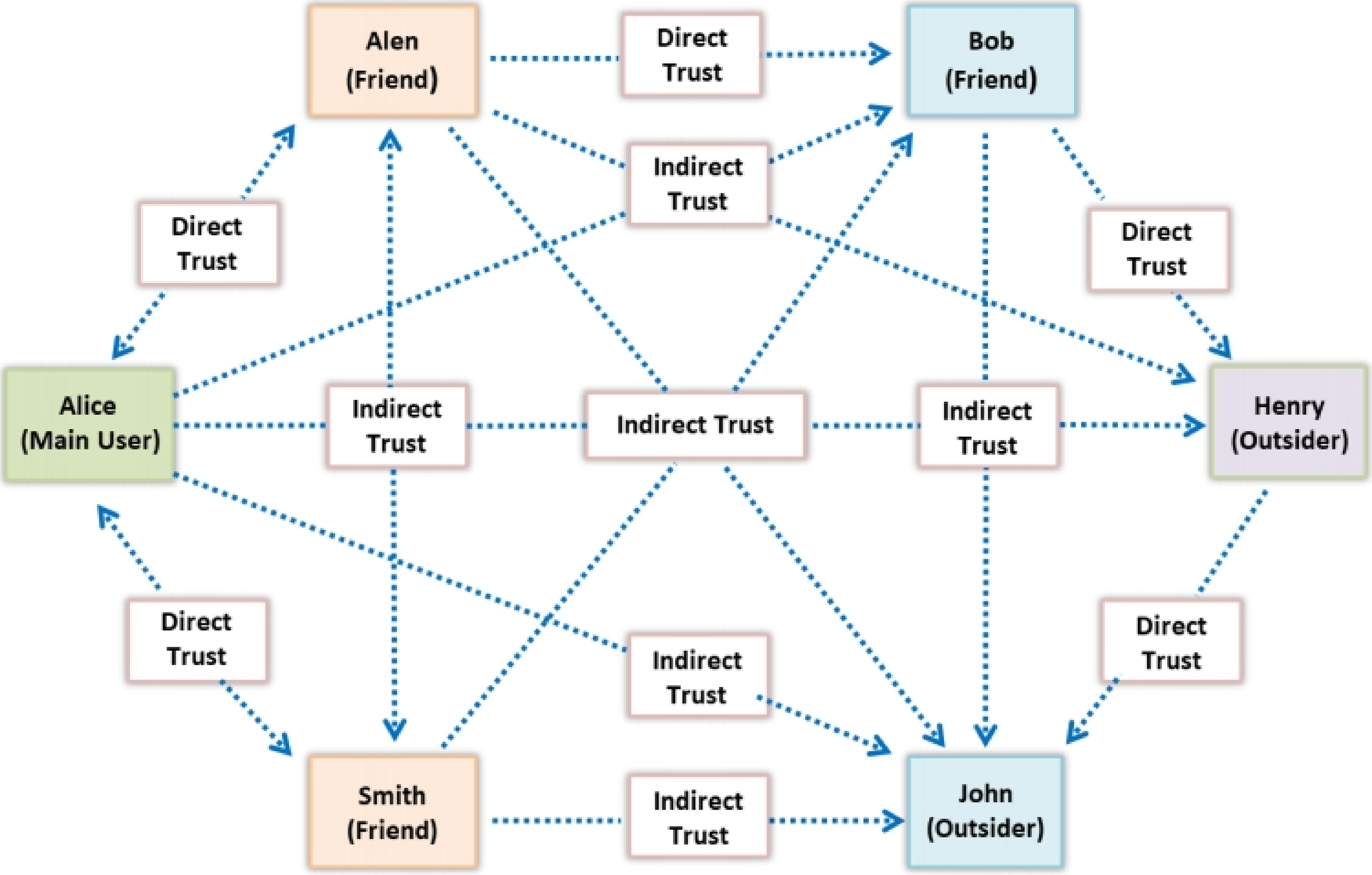
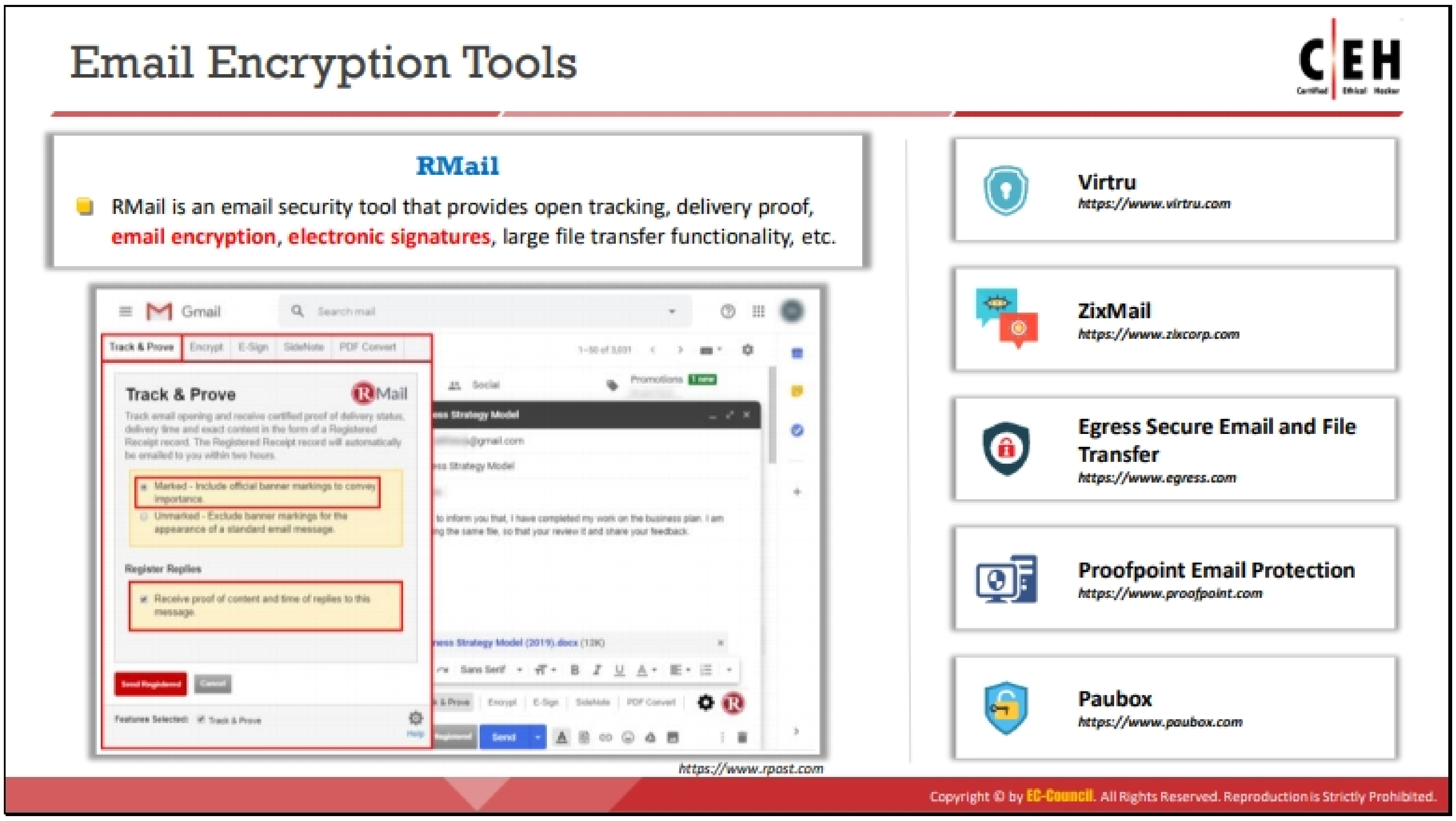

Alen
(Friend)
Direct Trust
Bob
(Friend)
Indirect Trust
Direct Trust
Direct Trust
Alice
(Main User)
Indirect Trust
Indirect Trust
Indirect Trust
Henry
(Outsider)
Direct Trust
Indirect Trust
Direct Trust
Smith
(Friend)
Indirect Trust
John
(Outsider)

Figure 20.32: Working of WOT



Email Encryption Tools

RMail

- RMail is an email security tool that provides open tracking, delivery proof, **email encryption**, **electronic signatures**, large file transfer functionality, etc.

https://www.rpost.com

Virtru
https://www.virtru.com

ZixMail
https://www.zixcorp.com

Egress Secure Email and File Transfer
https://www.egress.com

Proofpoint Email Protection
https://www.proofpoint.com

Paubox
https://www.paubox.com

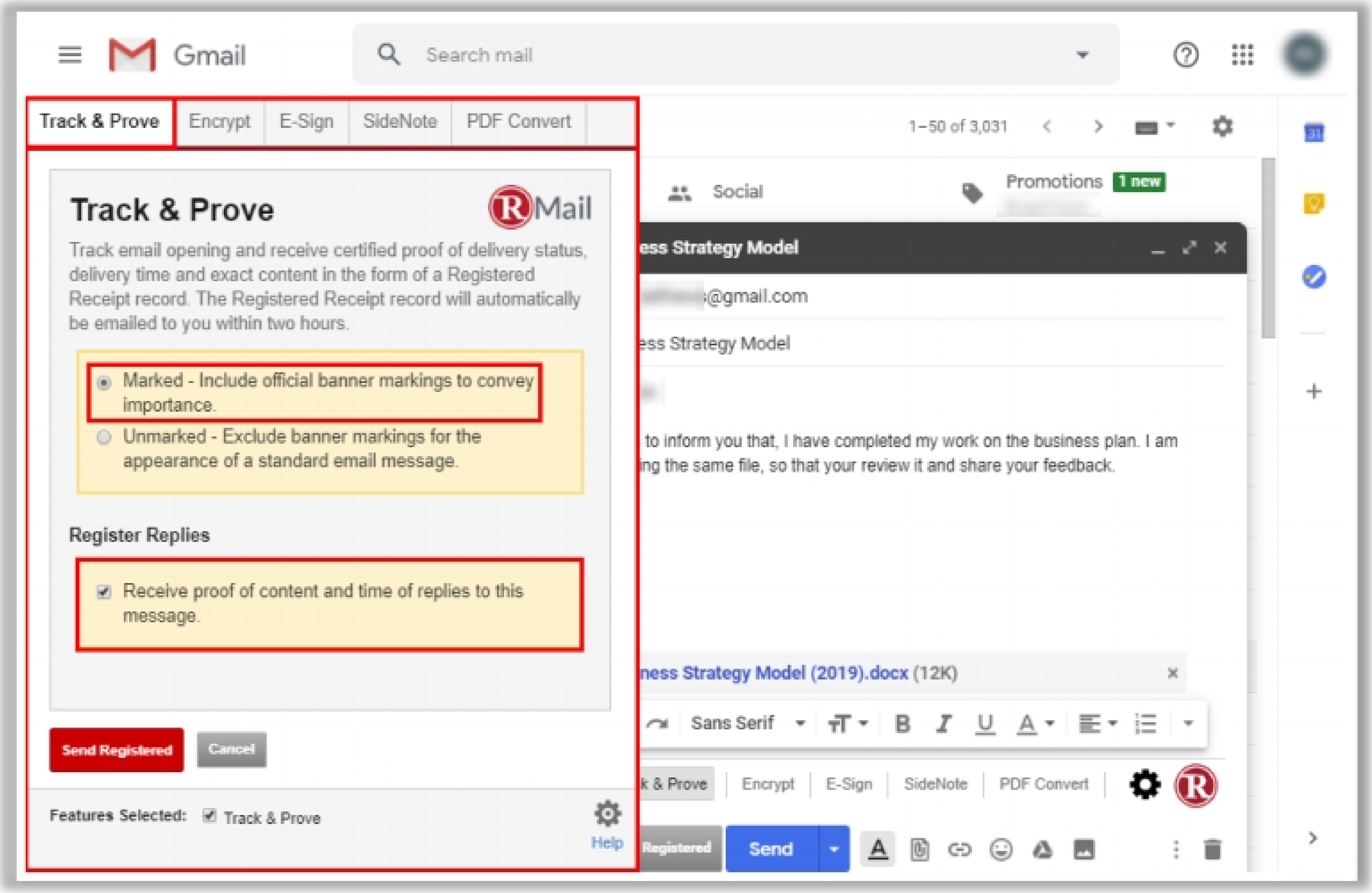
Email Encryption Tools

Some important email encryption tools used to secure email messages are as follows:

- RMail

  Source: *https://www.rpost.com*

  RMail is an email security tool that provides open tracking, delivery proof, email encryption, electronic signatures, large file transfer functionality, etc. RMail works seamlessly with users' existing email platforms, including Microsoft Outlook, Gmail, etc. Using this tool, you can encrypt sensitive emails and attachments for security or legal compliance.


Gmail
Search mail
Track & Prove
Encrypt
E-Sign
SideNote
PDF Convert
1–50 of 3,031
Social
Promotions
1 new
RMail
Send Registered
Cancel
Help
Sans Serif
Send
Registered

Track & Prove

Track email opening and receive certified proof of delivery status, delivery time and exact content in the form of a Registered Receipt record. The Registered Receipt record will automatically be emailed to you within two hours.

● Marked - Include official banner markings to convey importance.

○ Unmarked - Exclude banner markings for the appearance of a standard email message.

Register Replies

☑ Receive proof of content and time of replies to this message.

Features Selected: ☑ Track & Prove

ess Strategy Model

@gmail.com

ess Strategy Model

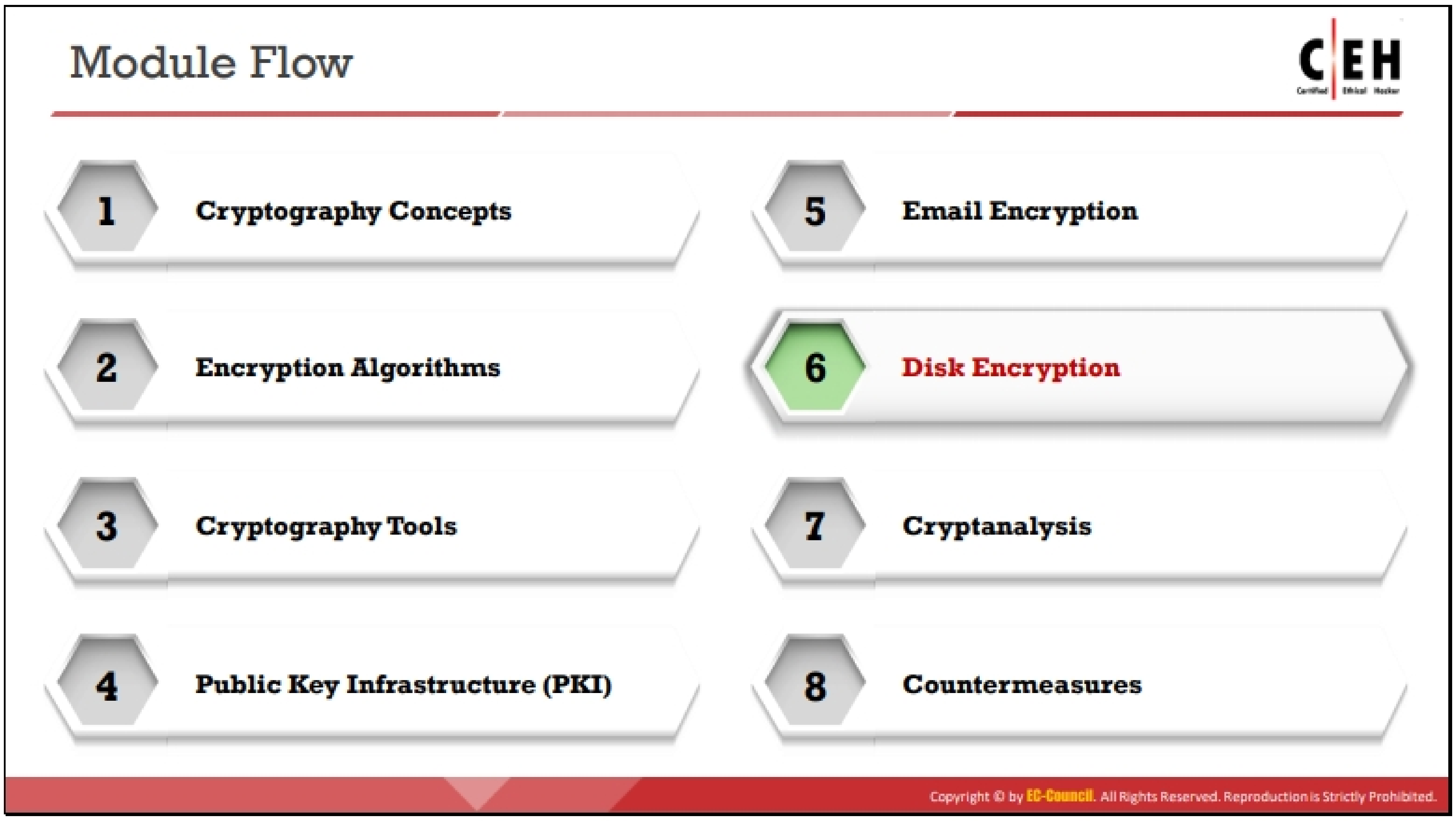
to inform you that, I have completed my work on the business plan. I am
ing the same file, so that your review it and share your feedback.

ness Strategy Model (2019).docx (12K)

Figure 20.33: Screenshot of RMail

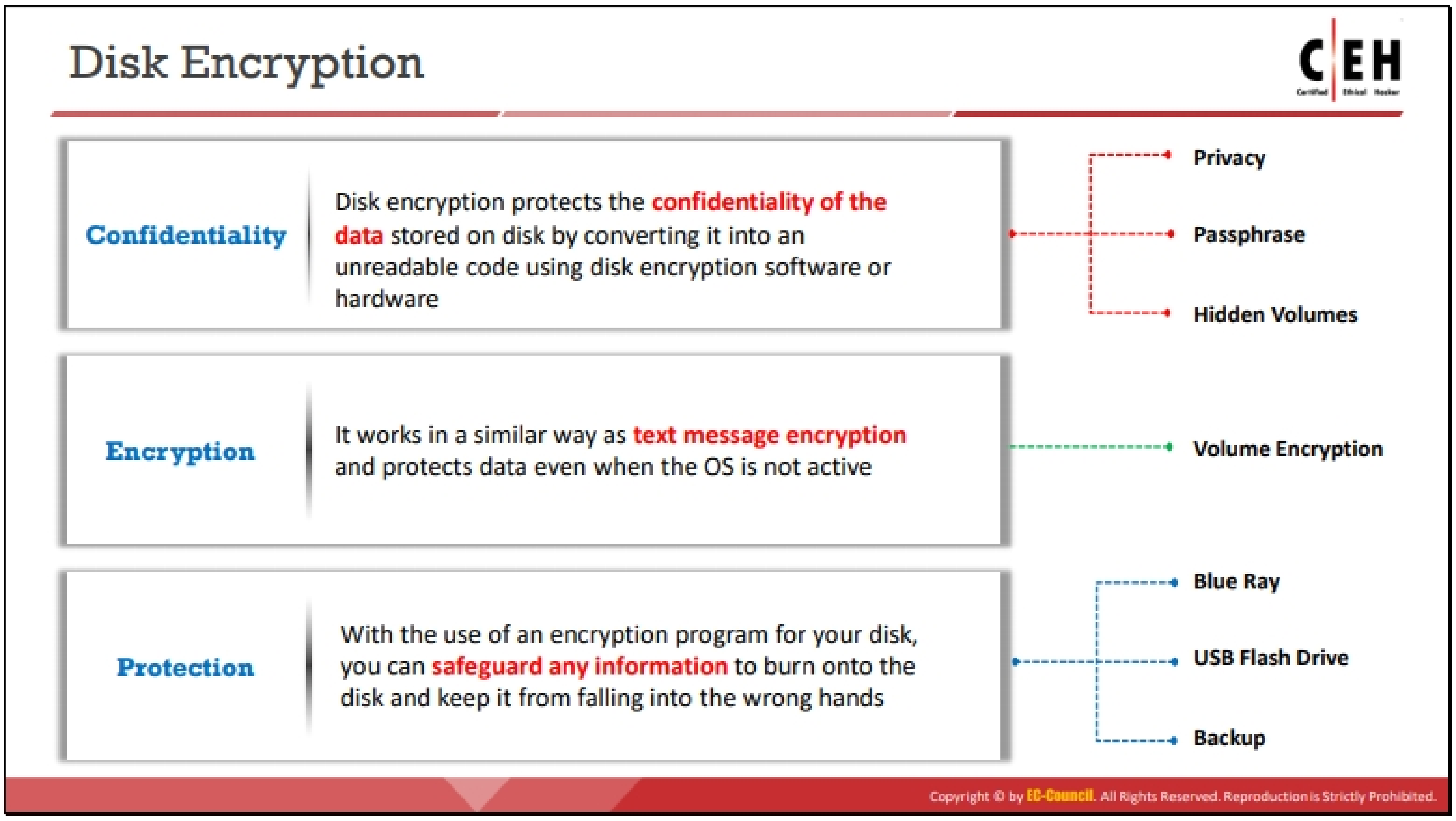
Some additional email encryption tools are as follows:

- Virtru (*https://www.virtru.com*)
- ZixMail (*https://www.zixcorp.com*)
- Egress Secure Email and File Transfer (*https://www.egress.com*)
- Proofpoint Email Protection (*https://www.proofpoint.com*)
- Paubox (*https://www.paubox.com*)



Module Flow

1 **Cryptography Concepts**
2 **Encryption Algorithms**
3 **Cryptography Tools**
4 **Public Key Infrastructure (PKI)**
5 **Email Encryption**
6 **Disk Encryption**
7 **Cryptanalysis**
8 **Countermeasures**



Disk Encryption

Confidentiality | Disk encryption protects the **confidentiality of the data** stored on disk by converting it into an unreadable code using disk encryption software or hardware

- Privacy
- Passphrase
- Hidden Volumes

Encryption | It works in a similar way as **text message encryption** and protects data even when the OS is not active

- Volume Encryption

Protection | With the use of an encryption program for your disk, you can **safeguard any information** to burn onto the disk and keep it from falling into the wrong hands

- Blue Ray
- USB Flash Drive
- Backup

Disk Encryption

Disk encryption encrypts every bit of data stored on a disk or a disk volume, thus preventing illegal access to data storage. This section deals with disk encryption concepts and various disk encryption tools.

Disk encryption is a technology that protects the confidentiality of the data stored on a disk by converting it into an unreadable code using disk encryption software or hardware, thus preventing unauthorized users from accessing it. Disk encryption provides confidentiality and privacy using passphrases and hidden volumes.

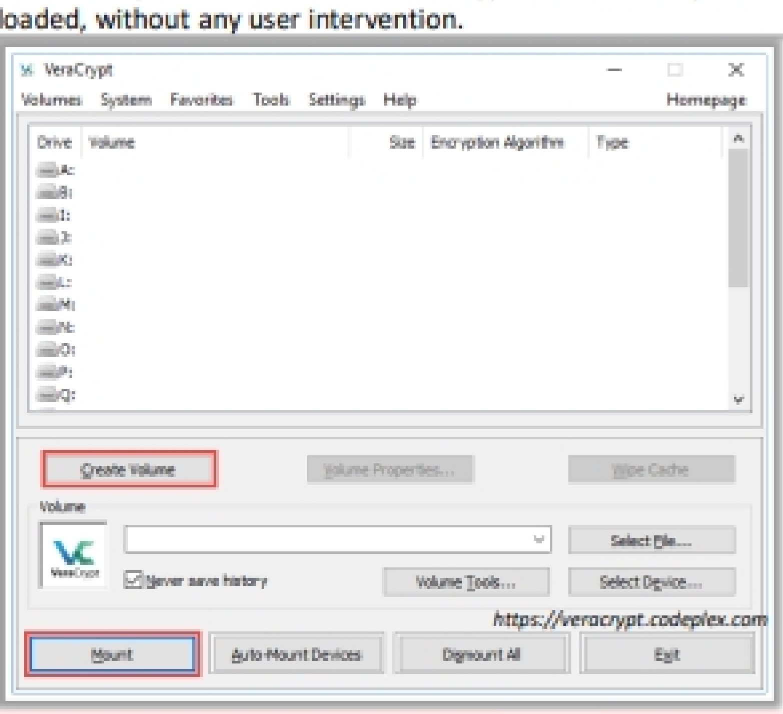
Disk encryption works similarly to text-message encryption and protects data even when the OS is not active. By using an encryption program for one's disk (Blue Ray, DVD, USB flash drive, external HDD, backup), one can safeguard any or all information on the disk and prevent it from falling into the wrong hands. Disk-encryption software scrambles the information on the disk into an illegible code. It is only after decryption of the disk information that one can read and use it.

Disk encryption is useful when the user needs to physically send sensitive information. In addition, disk encryption can protect the real-time exchange of information from compromising threats. When users exchange encrypted information, the chances of compromising the information are minimized. The only way an attacker can access the information is by decrypting the message. Furthermore, encryption software installed on a user's system ensures the security of the system. Install encryption software on any system that holds valuable information or systems that are exposed to unlimited data transfer.

Disk Encryption Tools: VeraCrypt and Symantec Drive Encryption

VeraCrypt

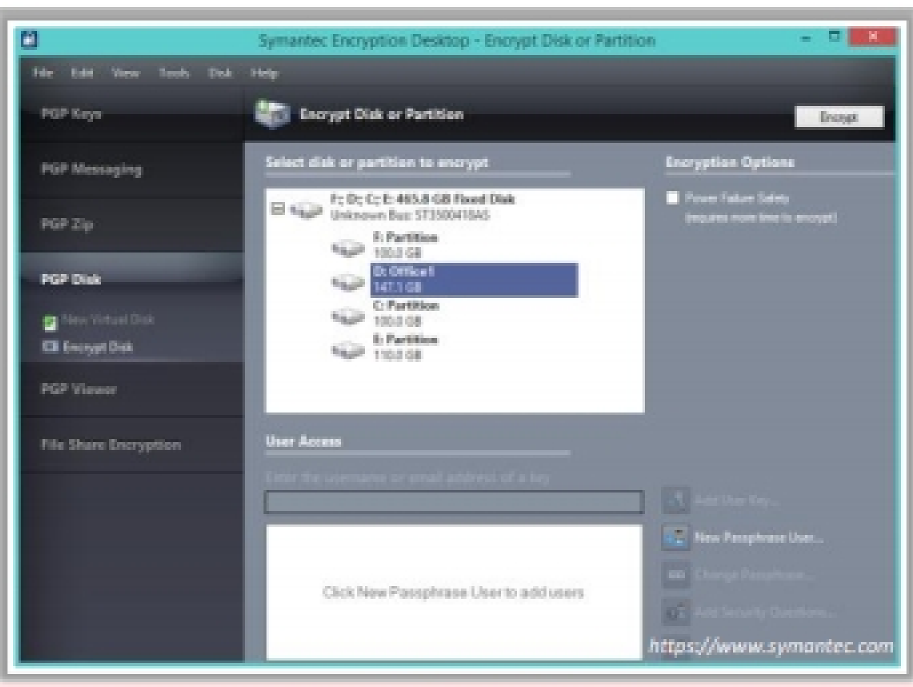
- VeraCrypt is a software for establishing and maintaining an **on-the-fly-encrypted volume** (data storage device).
- On-the-fly encryption means that **data is automatically encrypted** immediately before it is saved and decrypted immediately after it is loaded, without any user intervention.


VeraCrypt
Volumes System Favorites Tools Settings Help Homepage
Drive Volume Size Encryption Algorithm Type
A: B: I: J: K: L: M: N: O: P: Q:
Create Volume
Volume Properties...
Wipe Cache
Volume
Select File...
Never save history
Volume Tools...
Select Device...
Mount
Auto-Mount Devices
Dismount All
Exit

https://veracrypt.codeplex.com

Symantec Drive Encryption

- Symantec Drive Encryption provides **full disk encryption** for all data (user files, swap files, system files, etc.) on desktops, laptops, and removable media

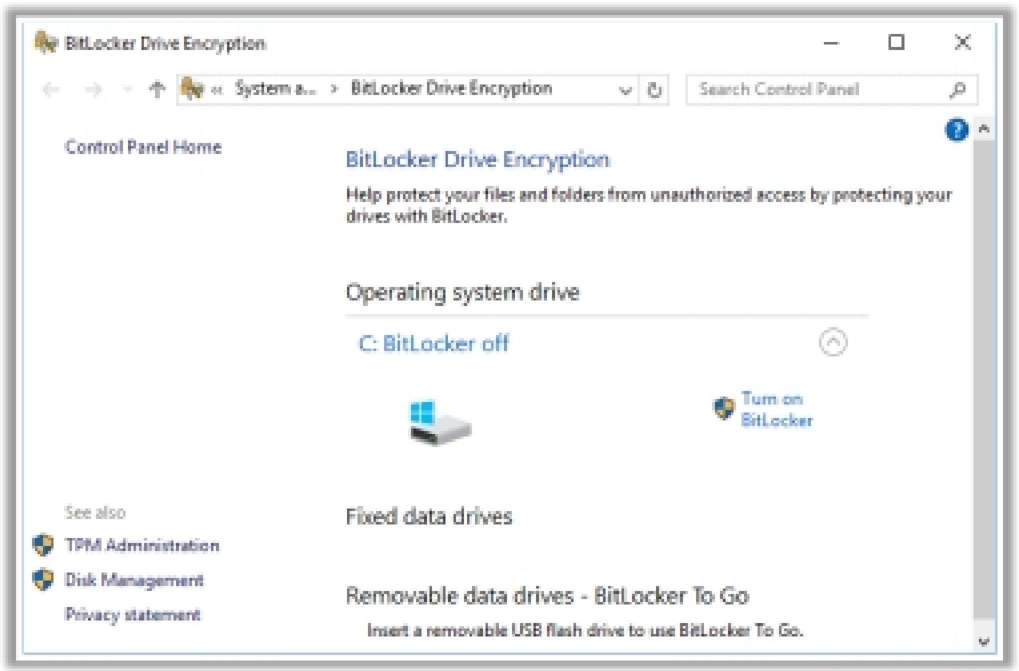

Symantec Encryption Desktop - Encrypt Disk or Partition
File Edit View Tools Disk Help
PGP Keys
PGP Messaging
PGP Zip
PGP Disk
New Virtual Disk
Encrypt Disk
PGP Viewer
File Share Encryption
Encrypt Disk or Partition
Encrypt
Select disk or partition to encrypt
Encryption Options
Power Failure Safety
User Access
Click New Passphrase User to add users

https://www.symantec.com

Disk Encryption Tools

BitLocker Drive Encryption

- BitLocker provides offline data and operating system protection for your computer
- It helps protect your data from theft or unauthorized viewing by encrypting the entire Windows volume


BitLocker Drive Encryption
System a... > BitLocker Drive Encryption
Search Control Panel
Control Panel Home
See also
TPM Administration
Disk Management
Privacy statement
C: BitLocker off
Turn on BitLocker

BitLocker Drive Encryption

Help protect your files and folders from unauthorized access by protecting your drives with BitLocker.

Operating system drive

Fixed data drives

Removable data drives - BitLocker To Go

Insert a removable USB flash drive to use BitLocker To Go.

https://docs.microsoft.com

- **FinalCrypt**
  http://www.finalcrypt.org
- **Seqrite Encryption Manager**
  https://www.seqrite.com
- **FileVault**
  https://support.apple.com
- **Gillsoft Full Disk Encryption**
  http://www.gillsoft.com
- **Rohos Disk Encryption**
  http://www.rohos.com

Disk Encryption Tools

The common goal of disk encryption tools is to encrypt a disk partition to provide confidentiality to the information stored on it. Some disk encryption tools are discussed below.

- **VeraCrypt**

  Source: *https://www.veracrypt.fr*

VeraCrypt is a software for establishing and maintaining an on-the-fly-encrypted volume (data storage device). On-the-fly encryption means that data is automatically encrypted just before it is saved and decrypted just after it is loaded without any user intervention. No data stored on an encrypted volume can be read (decrypted) without using the correct password/keyfile(s) or correct encryption keys. The entire file system is encrypted (e.g., file names, folder names, free space, metadata, etc.).

Files can be copied to and from a mounted VeraCrypt volume just like they are copied to/from any normal disk (e.g., by simple drag-and-drop operations). Files are automatically decrypted on the fly (in memory/RAM) while they are read or copied from an encrypted VeraCrypt volume. Similarly, files that are written or copied to the VeraCrypt volume are automatically encrypted on the fly (just before they are written to the disk) in RAM.

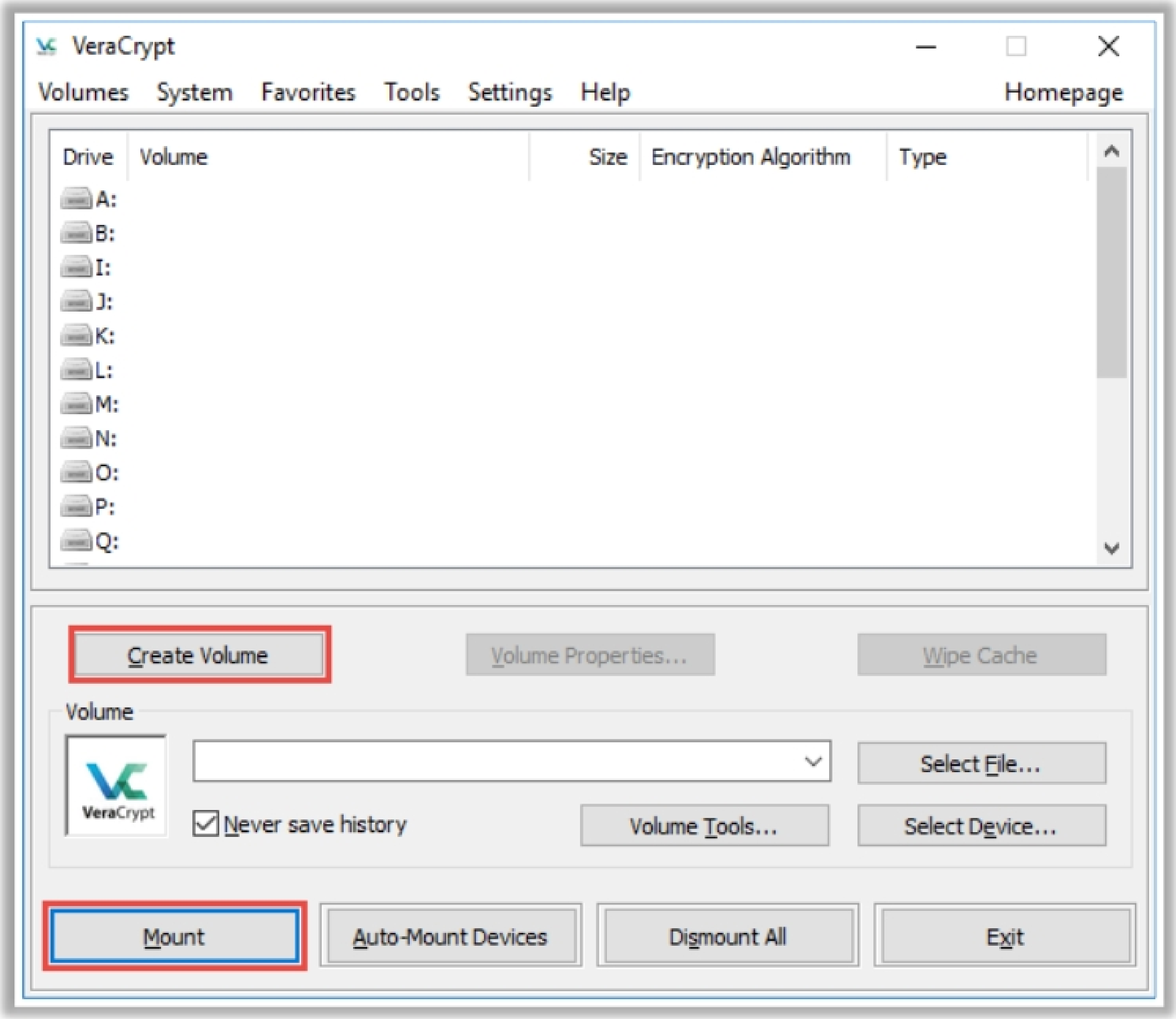

VeraCrypt
Volumes
System
Favorites
Tools
Settings
Help
Homepage
Drive
Volume
Size
Encryption Algorithm
Type
A:
B:
I:
J:
K:
L:
M:
N:
O:
P:
Q:
Create Volume
Volume Properties...
Wipe Cache
Volume
Select File...
☑ Never save history
Volume Tools...
Select Device...
Mount
Auto-Mount Devices
Dismount All
Exit

Figure 20.34: Screenshot of VeraCrypt

- **Symantec Drive Encryption**

  Source: *https://www.symantec.com*

  Symantec Drive Encryption (formerly PGP Whole Disk Encryption) provides organizations with complete, transparent drive encryption for all data (user files, swap files, system files, hidden files, etc.) on laptops, desktops, and removable media. It protects data from unauthorized access, thereby providing strong security for intellectual property as well as customer and partner data. Symantec Encryption Management Server can centrally manage the protected systems, as it simplifies deployment, policy creation, distribution, and reporting.

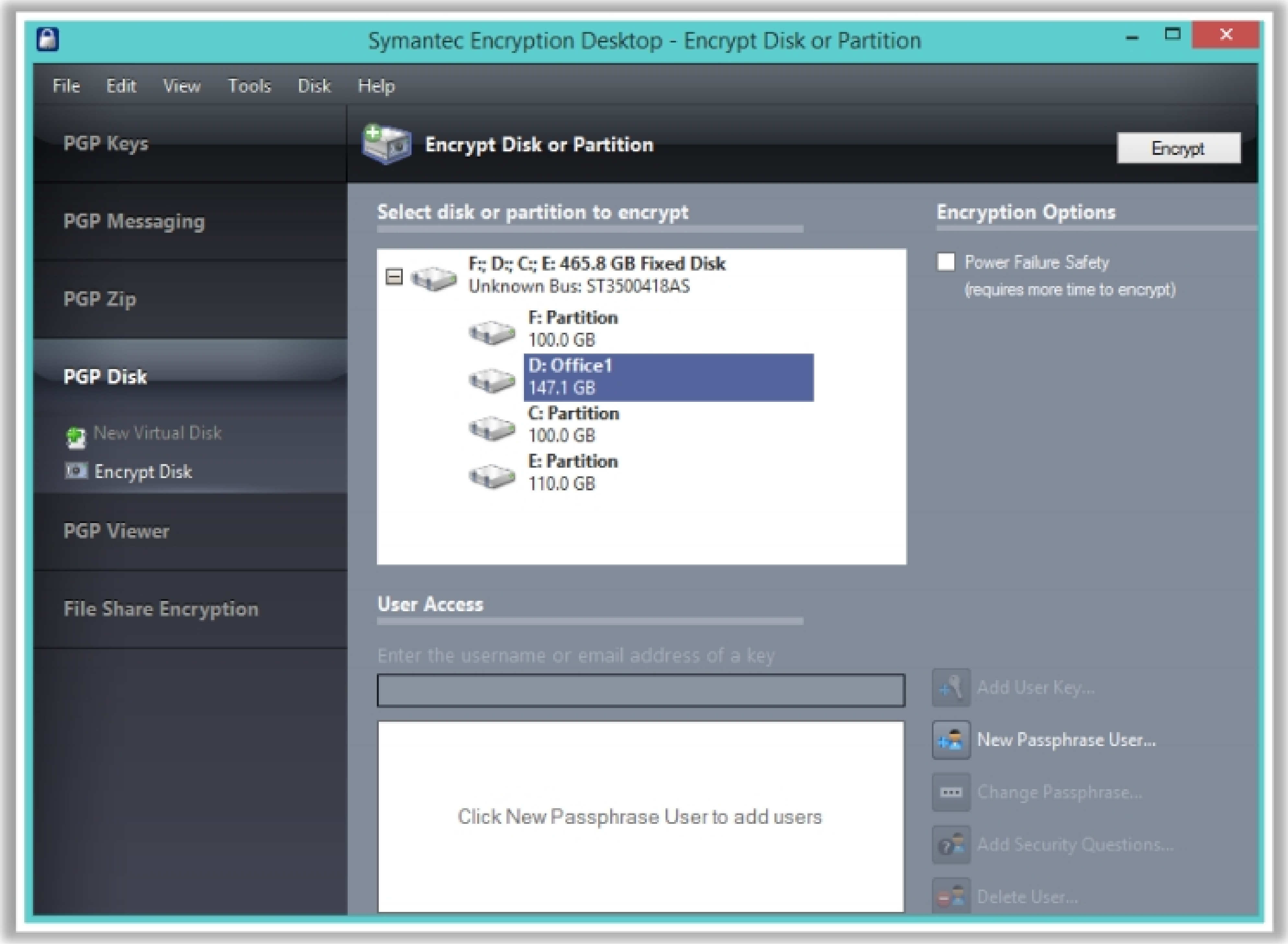
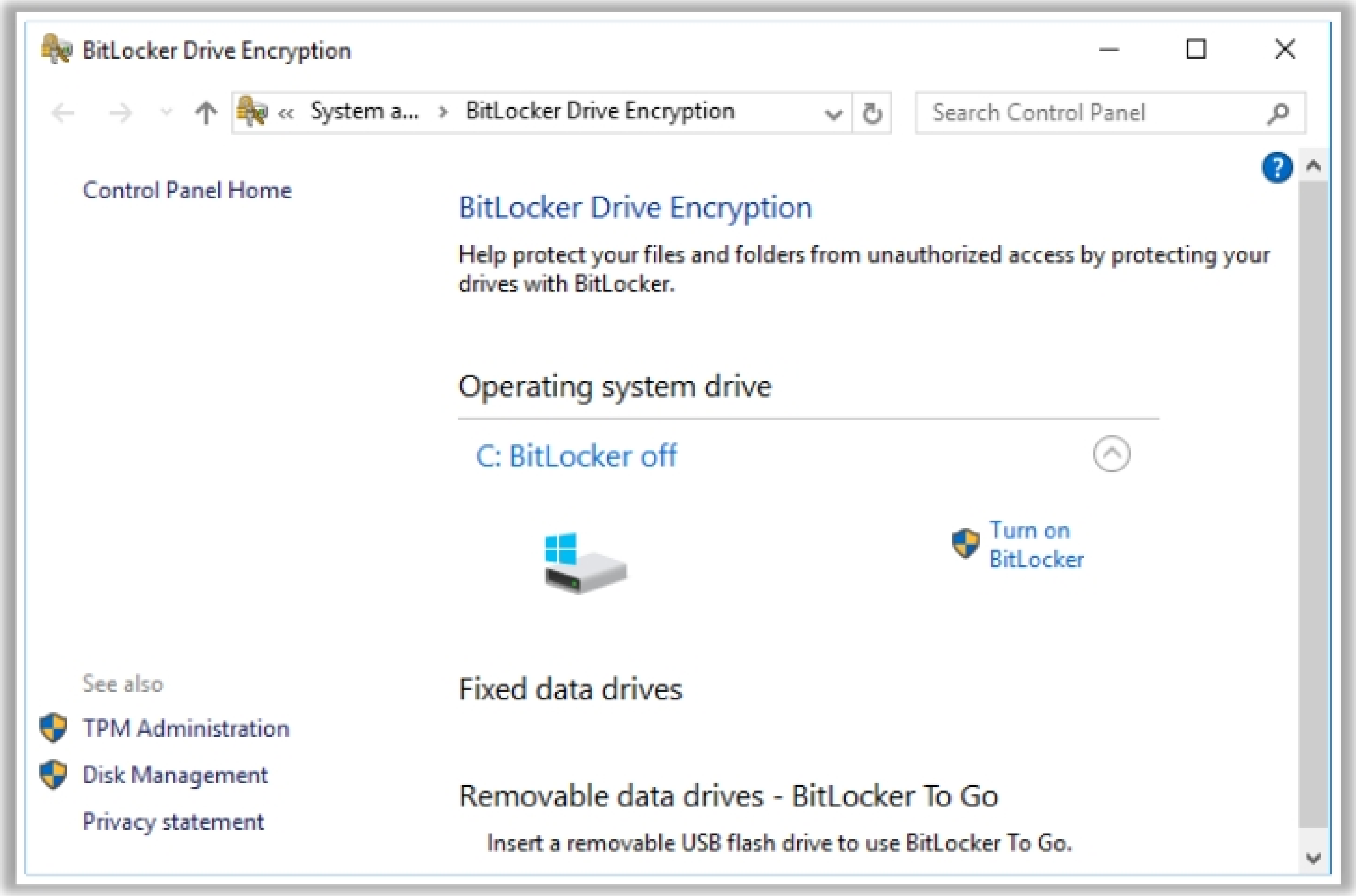

Symantec Encryption Desktop - Encrypt Disk or Partition
File
Edit
View
Tools
Disk
Help
PGP Keys
PGP Messaging
PGP Zip
PGP Disk
New Virtual Disk
Encrypt Disk
PGP Viewer
File Share Encryption
Encrypt Disk or Partition
Encrypt
Select disk or partition to encrypt
F:; D:; C:; E: 465.8 GB Fixed Disk
Unknown Bus: ST3500418AS
F: Partition
100.0 GB
D: Office1
147.1 GB
C: Partition
100.0 GB
E: Partition
110.0 GB
Encryption Options
☐ Power Failure Safety
(requires more time to encrypt)
User Access
Enter the username or email address of a key
Click New Passphrase User to add users
Add User Key...
New Passphrase User...
Change Passphrase...
Add Security Questions...
Delete User...

FIGURE 20.35: Screenshot of Symantec Drive Encryption

- **BitLocker Drive Encryption**

  Source: *https://docs.microsoft.com*

  BitLocker provides offline-data and OS protection for your computer. It helps ensure that data that is stored on a computer that is running Windows® is not revealed if the computer is tampered with when the installed OS is offline. BitLocker uses a microchip that is called a Trusted Platform Module (TPM) to provide enhanced protection for your data and to preserve early boot-component integrity. The TPM can help protect your data from theft or unauthorized access by encrypting the entire Windows volume.


BitLocker Drive Encryption
System a... > BitLocker Drive Encryption
Search Control Panel
Control Panel Home
See also
TPM Administration
Disk Management
Privacy statement
Turn on BitLocker

BitLocker Drive Encryption

Help protect your files and folders from unauthorized access by protecting your drives with BitLocker.

Operating system drive

C: BitLocker off

Fixed data drives

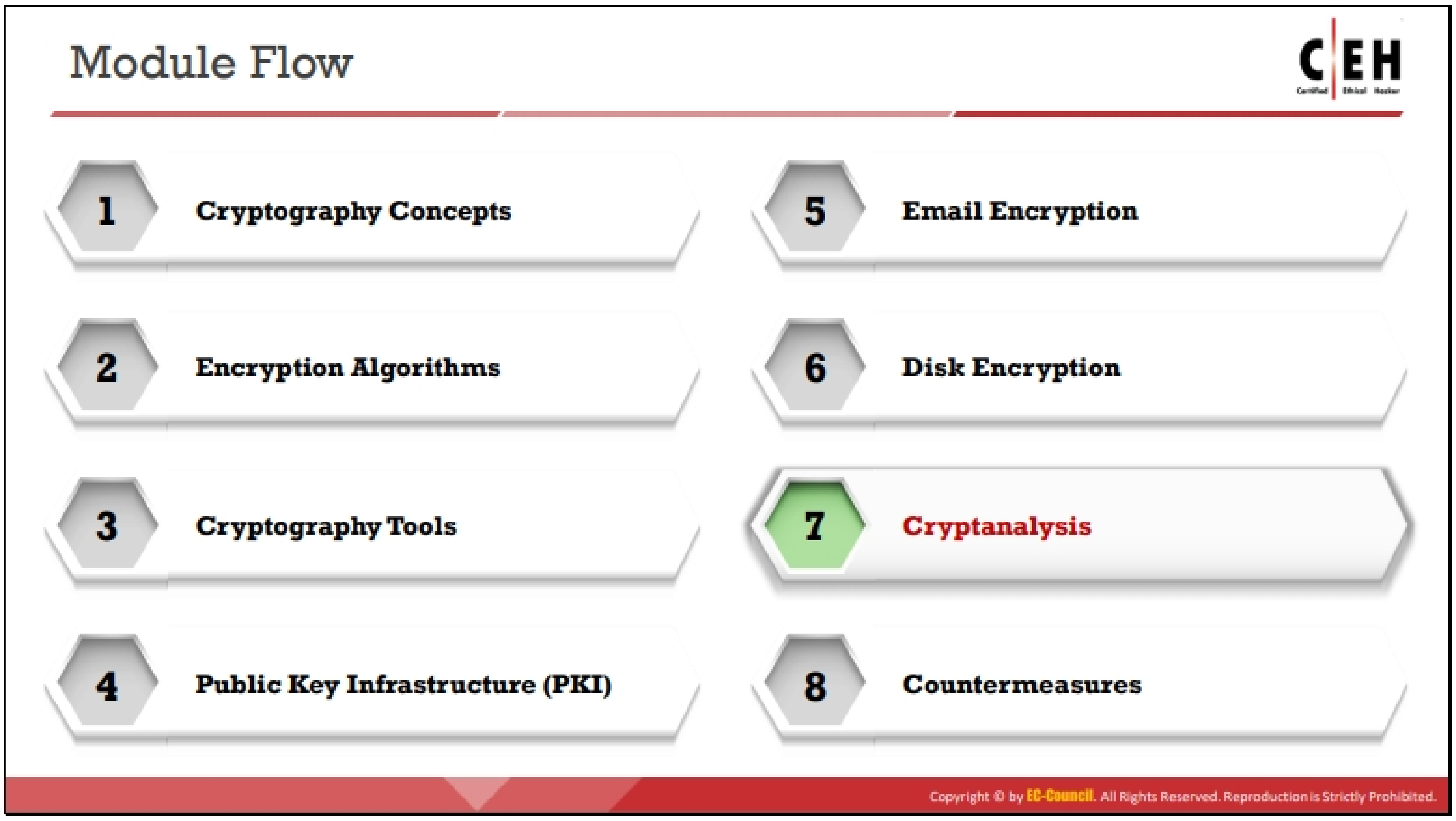
Removable data drives - BitLocker To Go

Insert a removable USB flash drive to use BitLocker To Go.

Figure 20.36: Screenshot of BitLocker Drive Encryption

Some additional disk encryption tools are as follows:

- FinalCrypt (*http://www.finalcrypt.org*)
- Seqrite Encryption Manager (*https://www.seqrite.com*)
- FileVault (*https://support.apple.com*)
- Gillsoft Full Disk Encryption (*http://www.gilisoft.com*)
- Rohos Disk Encryption (*http://www.rohos.com*)



Module Flow

1 **Cryptography Concepts**

2 **Encryption Algorithms**

3 **Cryptography Tools**

4 **Public Key Infrastructure (PKI)**

5 **Email Encryption**

6 **Disk Encryption**

7 **Cryptanalysis**

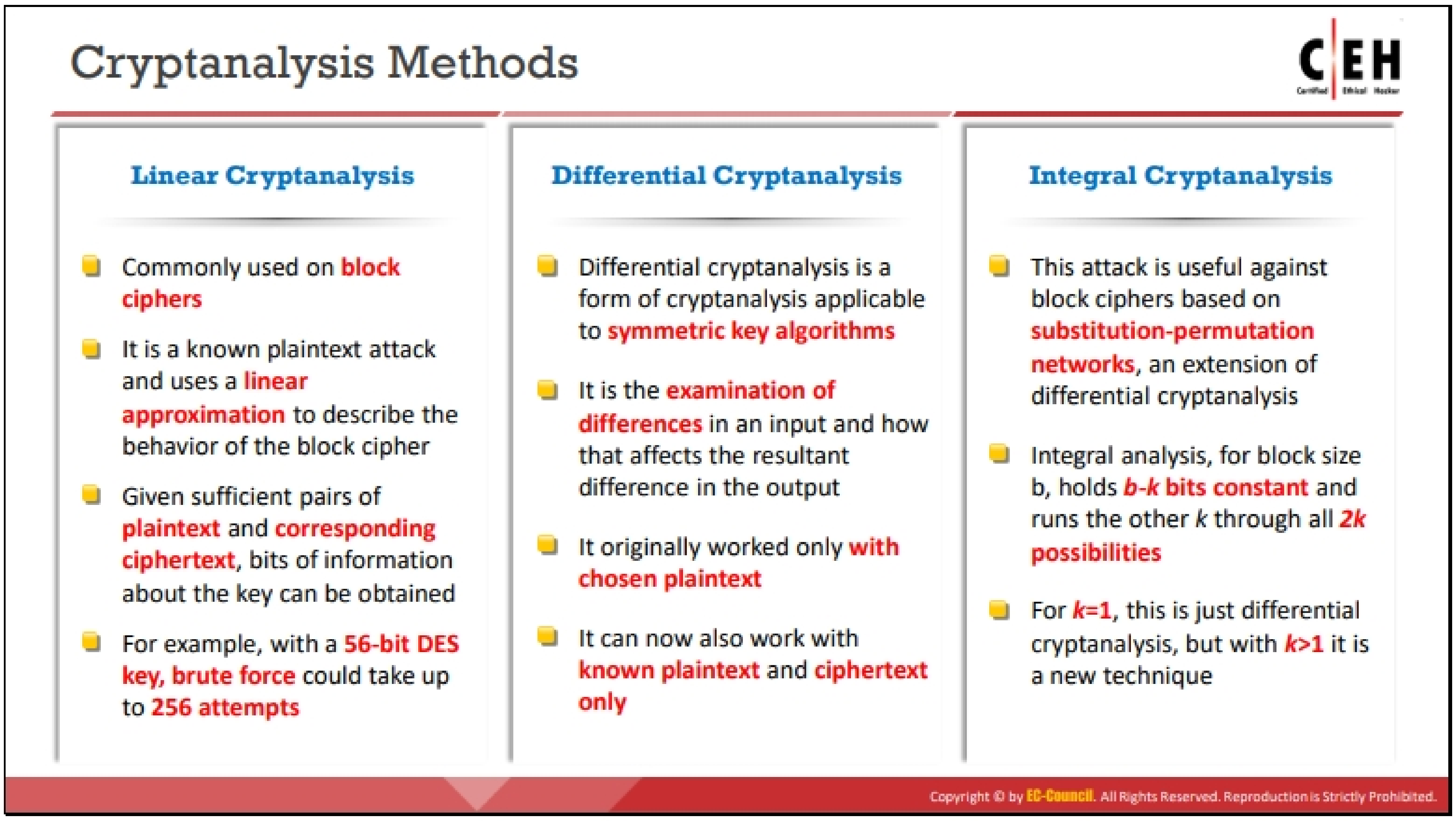
8 **Countermeasures**

Cryptanalysis

Attackers may implement various cryptography attacks to evade the security of a cryptographic system by exploiting vulnerabilities in code, ciphers, cryptographic protocols, or key management schemes. This process is known as cryptanalysis.

Cryptanalysis is the study of ciphers, ciphertext, or cryptosystems with the ability to identify vulnerabilities in them and thus extract plaintext from ciphertext even if the cryptographic key or algorithm used to encrypt the plaintext is unknown.

This section deals with various cryptography attacks that an attacker performs to compromise cryptographic systems as well as various cryptanalysis techniques and tools that help in breaching cryptographic security.



Cryptanalysis Methods

Linear Cryptanalysis

- Commonly used on **block ciphers**
- It is a known plaintext attack and uses a **linear approximation** to describe the behavior of the block cipher
- Given sufficient pairs of **plaintext** and **corresponding ciphertext**, bits of information about the key can be obtained
- For example, with a **56-bit DES key, brute force** could take up to **256 attempts**

Differential Cryptanalysis

- Differential cryptanalysis is a form of cryptanalysis applicable to **symmetric key algorithms**
- It is the **examination of differences** in an input and how that affects the resultant difference in the output
- It originally worked only **with chosen plaintext**
- It can now also work with **known plaintext** and **ciphertext only**

Integral Cryptanalysis

- This attack is useful against block ciphers based on **substitution-permutation networks**, an extension of differential cryptanalysis
- Integral analysis, for block size b, holds ***b-k* bits constant** and runs the other *k* through all **2k possibilities**
- For ***k*=1**, this is just differential cryptanalysis, but with ***k*>1** it is a new technique

Cryptanalysis Methods

- **Linear Cryptanalysis**

  Linear cryptanalysis is based on finding affine approximations to the action of a cipher. It is commonly used on block ciphers. This technique was invented by Mitsarue Matsui. It is a known plaintext attack and uses a linear approximation to describe the behavior of the block cipher. Given sufficient pairs of plaintext and corresponding ciphertext, bits of information about the key can be obtained. Obviously, the more pairs of plaintext and ciphertext one has, the greater are the chances of success.

  Remember that cryptanalysis is an attempt to crack cryptography. For example, with the 56-bit data encryption standard (DES), **key brute-forcing could take up to 256 attempts**. Linear cryptanalysis require $\mathbf{2^{43}}$ known plaintexts. This is better than brute-forcing but is still impractical in most situations. The math may be a bit complex for novice cryptographers, but let us look at the basics of it.

  With this method, a linear equation expresses the equality of two expressions, which consists of XORed binary variables. For example, the following equation XORs the sum of the first and third plaintext bits, and the first ciphertext bit is equal to the second bit of the key:

  $$P_1 \oplus P_3 \oplus C1 = K_2$$

  You can use this method to slowly re-create the key that was used.

  After doing this for each bit, you will have an equation of the form

  $$P_{i1} \oplus P_{i2} \oplus \dots \oplus C_{j1} \oplus C_{j2} \oplus \dots = K_{k1} \oplus K_{k2} \oplus \dots$$

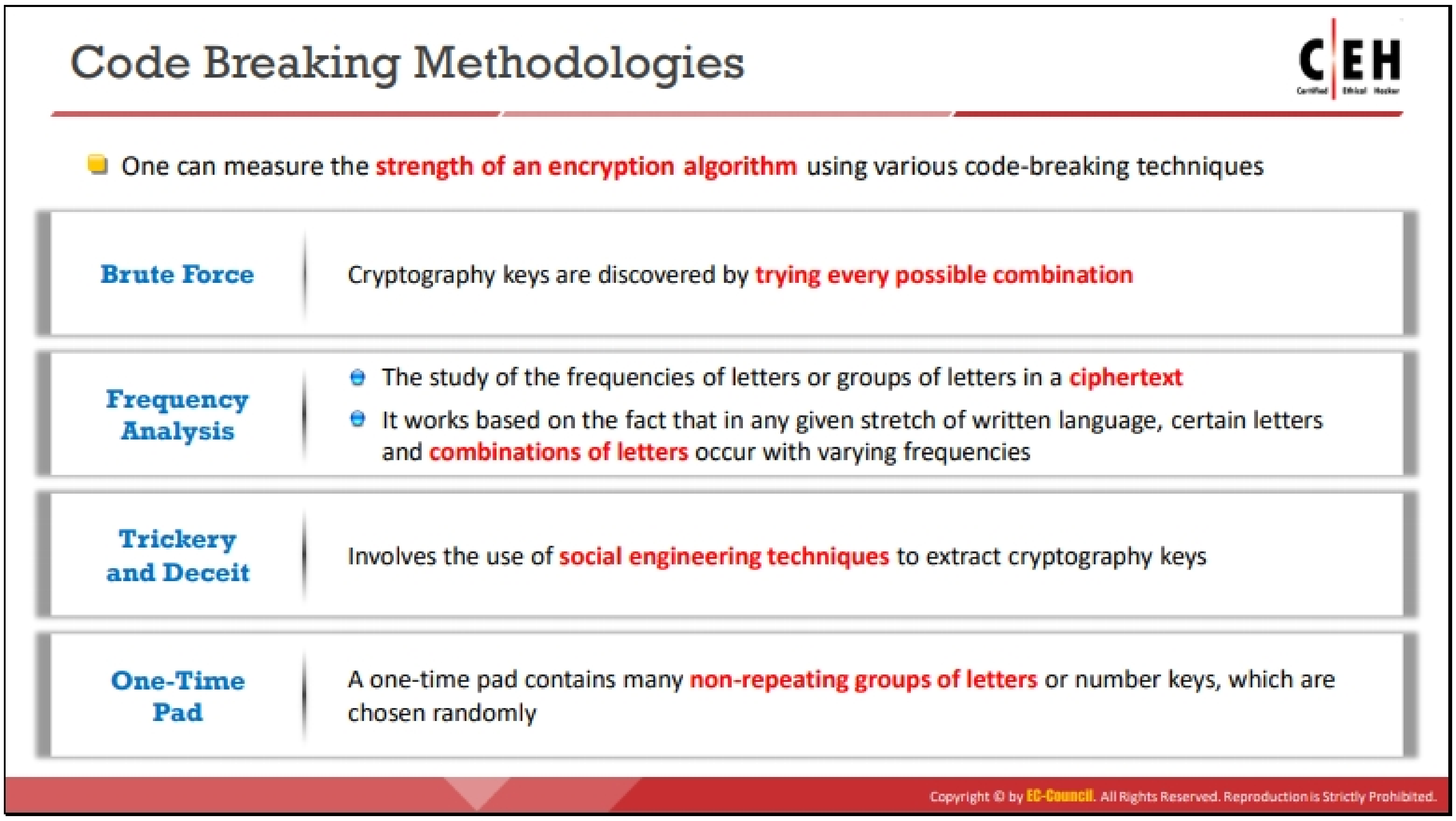
You can then use **Matsui's Algorithm 2**, using known plaintext-ciphertext pairs, to guess the values of the key bits involved in the approximation. For each set of values of the key bits on the right-hand side (referred to as a partial key), count how many times the approximation holds true over all the known plaintext-ciphertext pairs; call this count T. The partial key whose T has the greatest absolute difference from half the number of plaintext-ciphertext pairs is designated as the most likely set of values for those key bits.

- **Differential Cryptanalysis**

  Differential cryptanalysis is a form of cryptanalysis applicable to symmetric-key algorithms. It was invented by Eli Biham and Adi Shamir. Essentially, it is the examination of differences in input and how that affects the resultant difference in the output. It originally worked only with chosen plaintext. It can also work with known **plaintext** and **ciphertext**.

- **Integral Cryptanalysis**

  Integral cryptanalysis was first described by Lars Knudsen. This attack is particularly useful against block ciphers based on substitution-permutation networks as an extension of differential cryptanalysis. The differential analysis looks at pairs of inputs that differ in only one bit position, with all other bits being identical. Integral analysis for block size b holds b-k bits constant and runs the other k bits through all 2k possibilities. For k = 1, this is just differential cryptanalysis, but with k > 1, it is a new technique.



Code Breaking Methodologies

- One can measure the **strength of an encryption algorithm** using various code-breaking techniques

Brute Force	Cryptography keys are discovered by **trying every possible combination**
Frequency Analysis	The study of the frequencies of letters or groups of letters in a **ciphertext** It works based on the fact that in any given stretch of written language, certain letters and **combinations of letters** occur with varying frequencies
Trickery and Deceit	Involves the use of **social engineering techniques** to extract cryptography keys
One-Time Pad	A one-time pad contains many **non-repeating groups of letters** or number keys, which are chosen randomly

Code Breaking Methodologies

One can measure the strength of an encryption algorithm using various code breaking techniques, some of which are as follows:

- **Brute Force**

  Code breakers or cryptanalysts work to recover the plaintext of a message without knowing the required key in advance. They may first try to recover the key, or they may go after the message itself. A common cryptanalytic technique is a brute-force attack, or exhaustive search, in which the keys are determined by trying every possible combination of characters.

  The efficiency of a brute-force attack depends on the hardware configuration. The use of faster processors means that more keys will be tested per second. Cryptanalysts carried out a successful brute-force attack on a DES encryption method, which effectively rendered DES obsolete.

- **Frequency Analysis**

  Frequency analysis is the study of the frequency of letters or groups of letters in a ciphertext. Frequency analysis of letters and words is another method used to crack ciphers. It works on the principle that, in any given stretch of written language, certain letters and combinations of letters occur with varying frequencies. This technique examines the number of times that a particular symbol appears in a ciphertext. For example, the letter "**e**" is a common letter in the English language. If the letter "**k**" appears commonly in a ciphertext, it can be reasonably concluded that "**k**" in the encrypted language is equivalent to "**e**" in English.

Encrypted source code is more vulnerable to these types of attacks because words such as “**#define**,” “**struct,**” “**else,**” and “**return**” are repeated frequently in code. Sophisticated cryptosystems are required to maintain the security of messages against frequency analysis.

- **Trickery and Deceit**

  Trickery and deceit require a high level of mathematical and cryptographic skills. It involves the use of social engineering techniques to extract cryptography keys.

  Example: It is fairly easy to decrypt an entire message if the user knows some of its content.

  An attacker can use social engineering techniques to trick or bribe someone to encrypt and send a known message, which, when intercepted, could then be easily decrypted using standard cryptanalysis techniques.

- **One-Time Pad**

  One can crack any cipher if provided with sufficient time and resources. However, there is an exception called a one-time pad, which users assume to be unbreakable even with infinite resources.

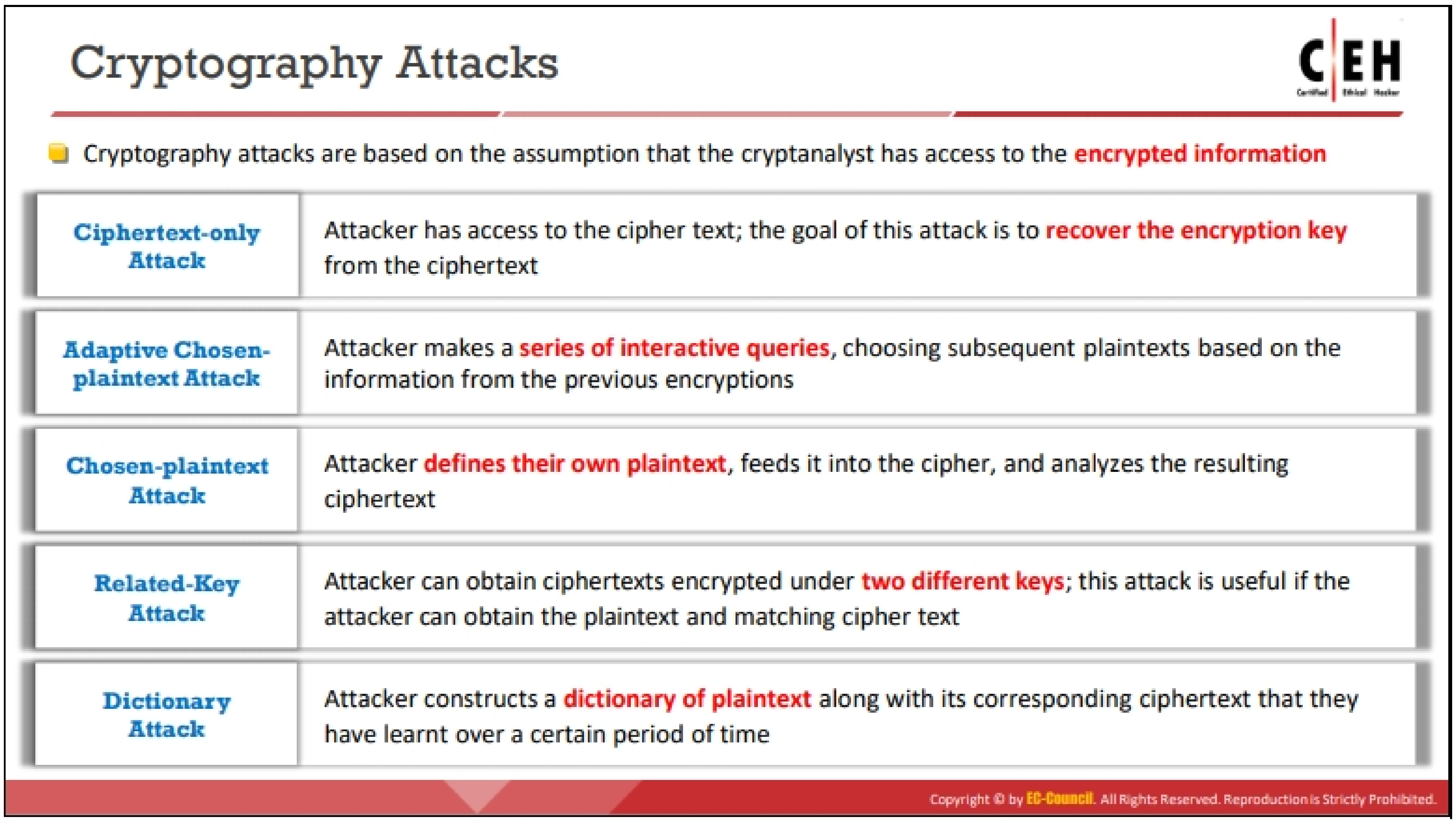
  A one-time pad mostly contains a non-repeating set of letters or numbers, which the system chooses randomly. The user writes them on small sheets of paper and then pastes them together in a pad.

  Example of One-time pad usage:

  The sender encrypts only one plaintext character using each key letter on the pad, and the receiver decrypts each letter of the ciphertext using an identical pad. Once the letter uses a page, he or she tears it off the pad and securely discards it; hence, the name one-time pad.

  Drawback:

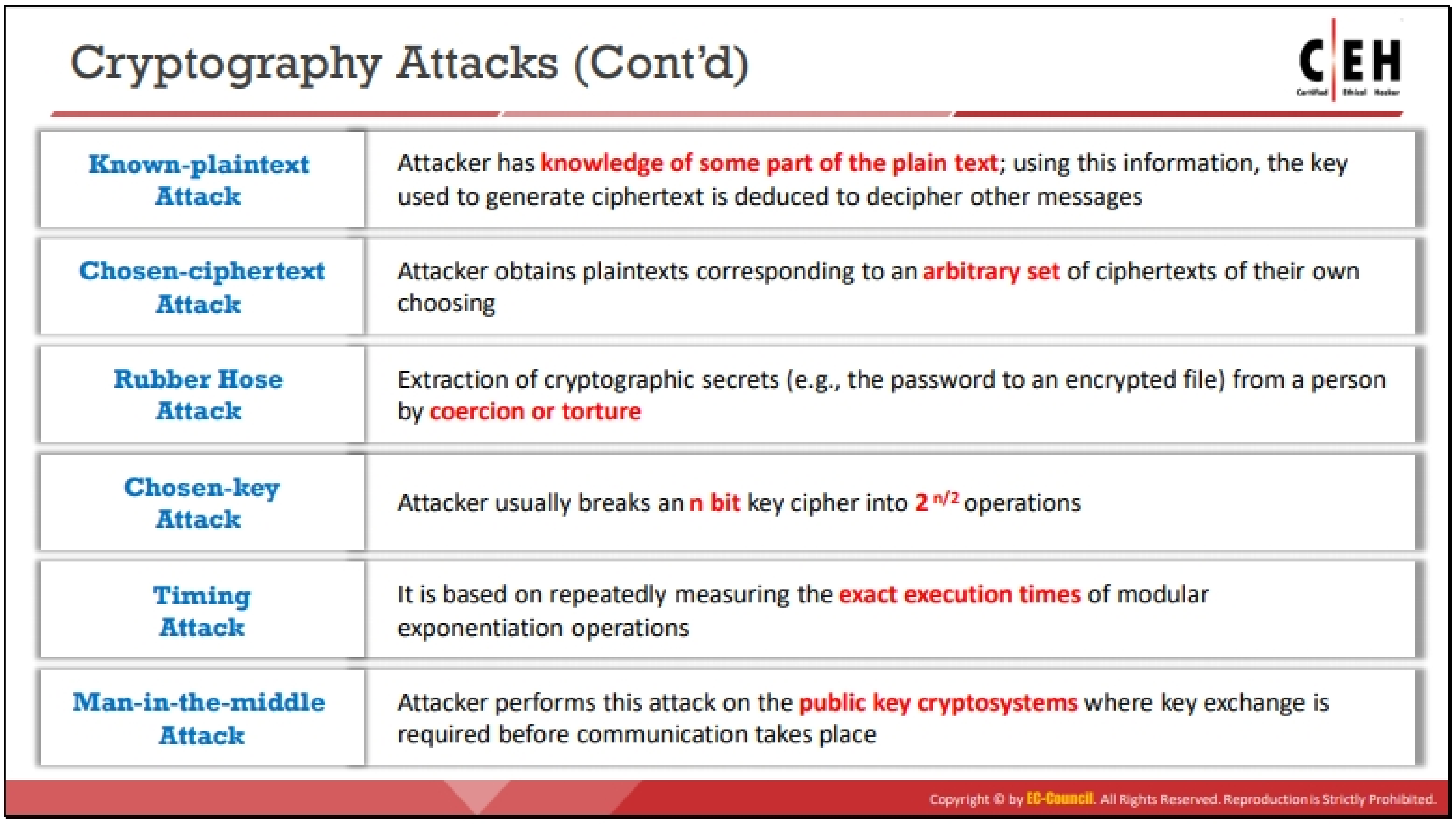
  The key length is the same as that of the message, thus making it impossible to encrypt and send large messages.



Cryptography Attacks

- Cryptography attacks are based on the assumption that the cryptanalyst has access to the **encrypted information**

Ciphertext-only Attack	Attacker has access to the cipher text; the goal of this attack is to **recover the encryption key** from the ciphertext
Adaptive Chosen-plaintext Attack	Attacker makes a **series of interactive queries**, choosing subsequent plaintexts based on the information from the previous encryptions
Chosen-plaintext Attack	Attacker **defines their own plaintext**, feeds it into the cipher, and analyzes the resulting ciphertext
Related-Key Attack	Attacker can obtain ciphertexts encrypted under **two different keys**; this attack is useful if the attacker can obtain the plaintext and matching cipher text
Dictionary Attack	Attacker constructs a **dictionary of plaintext** along with its corresponding ciphertext that they have learnt over a certain period of time



Cryptography Attacks (Cont'd)

Known-plaintext Attack	Attacker has **knowledge of some part of the plain text**; using this information, the key used to generate ciphertext is deduced to decipher other messages
Chosen-ciphertext Attack	Attacker obtains plaintexts corresponding to an **arbitrary set** of ciphertexts of their own choosing
Rubber Hose Attack	Extraction of cryptographic secrets (e.g., the password to an encrypted file) from a person by **coercion or torture**
Chosen-key Attack	Attacker usually breaks an **n bit** key cipher into $2^{n/2}$ operations
Timing Attack	It is based on repeatedly measuring the **exact execution times** of modular exponentiation operations
Man-in-the-middle Attack	Attacker performs this attack on the **public key cryptosystems** where key exchange is required before communication takes place

Cryptography Attacks

Attackers conduct cryptography attacks by assuming that the cryptanalyst has access to the encrypted information. A cryptography attack or cryptanalysis involves the study of various principles and methods of decrypting the ciphertext back to the plaintext without knowledge of the key.

The various types of cryptography attacks are as follows:

- **Ciphertext-only Attack**

  Ciphertext-only is less effective but much more likely for the attacker. The attacker only has access to a collection of ciphertexts. This is much more likely than known plaintext but is also the most difficult. The attack is completely successful if the corresponding plaintexts (or even better, the key) can be deduced. The ability to obtain any information at all about the underlying plaintext is still considered a success. So what does the attacker do with the ciphertexts he/she has accumulated? You can analyze them for patterns, trying to find something that would give you a hint as to the key that was used to crack them. Often, the result of this attack is just a partial break and not a complete break.

- **Adaptive Chosen-plaintext Attack**

  In this type of attack, an attacker has complete access to the plaintext message including its encryption, and he/she can also modify the content of the message by making a series of interactive queries, choosing subsequent plaintext blocks based on the information from the previous encryption queries and functions. To perform this attack, an attacker needs to interact with the encryption device.

- **Chosen-plaintext Attack**

  A chosen plaintext attack is a highly effective type of cryptanalysis attack. In this attack, the attacker obtains the ciphertexts corresponding to a set of plaintexts of his/her own choosing. This allows the attacker to attempt to derive the key used and thus decrypt other messages encrypted with that key. Basically, since the attacker knows the plaintext and the resultant ciphertext, he/she gains many insights into the key used. This technique can be difficult but is not impossible.

- **Related-Key Attack**

  The related-key attack is similar to the chosen plaintext attack, except that the attacker can obtain ciphertexts encrypted under two different keys. This is actually a very useful attack if you can obtain the plaintext and matching ciphertext. The attack requires that the differing keys be closely related, e.g., in a wireless environment where subsequent keys might be derived from previous keys. Then, while the keys are different, they are close. Much like the ciphertext-only attack, this type of attack is most likely only going to yield a partial break.

- **Dictionary Attack**

  In this attack, the attacker constructs a dictionary of plaintext along with its corresponding ciphertext that he/she has analyzed and obtained for a certain period of time. After building the dictionary, if the attacker obtains the ciphertext, he/she uses the already built dictionary to find the corresponding plaintext. Attackers use this technique to decrypt keys, passwords, passphrases, and ciphertext.

- **Known-plaintext Attack**

  In this attack, the only information available to the attacker is some plaintext blocks along with the corresponding ciphertext and algorithm used to encrypt and decrypt the text. Using this information, the key used to generate the ciphertext is deduced so as to decipher other messages. This attack works on block ciphers and is an example of linear cryptanalysis. The known plaintext blocks are generated using a series of intelligent guesses and logic, and not by accessing the plaintext over a channel.

- **Chosen-ciphertext Attack**

  The attacker obtains the plaintexts corresponding to an arbitrary set of ciphertexts of his own choosing. Using this information, the attacker tries to recover the key used to encrypt the plaintext. To perform this attack, the attacker must have access to the communication channel between the sender and the receiver.

  There are two variants of this attack:

  - Lunchtime or Midnight Attack: In this attack, the attacker can have access to the system for only a limited amount of time or can access only a few plaintext-ciphertext pairs.
  - Adaptive Chosen-ciphertext Attack: In this attack, the attacker selects a series of ciphertexts and then observes the resulting plaintext blocks.

- **Rubber Hose Attack**

  Attackers extract cryptographic secrets (e.g., the password to an encrypted file) from a person by coercion or torture. In general, people under pressure cannot maintain security, and they will reveal secrets or hidden information. Attackers torture victims to reveal secret keys or passwords used to encrypt the information.

- **Chosen-key Attack**

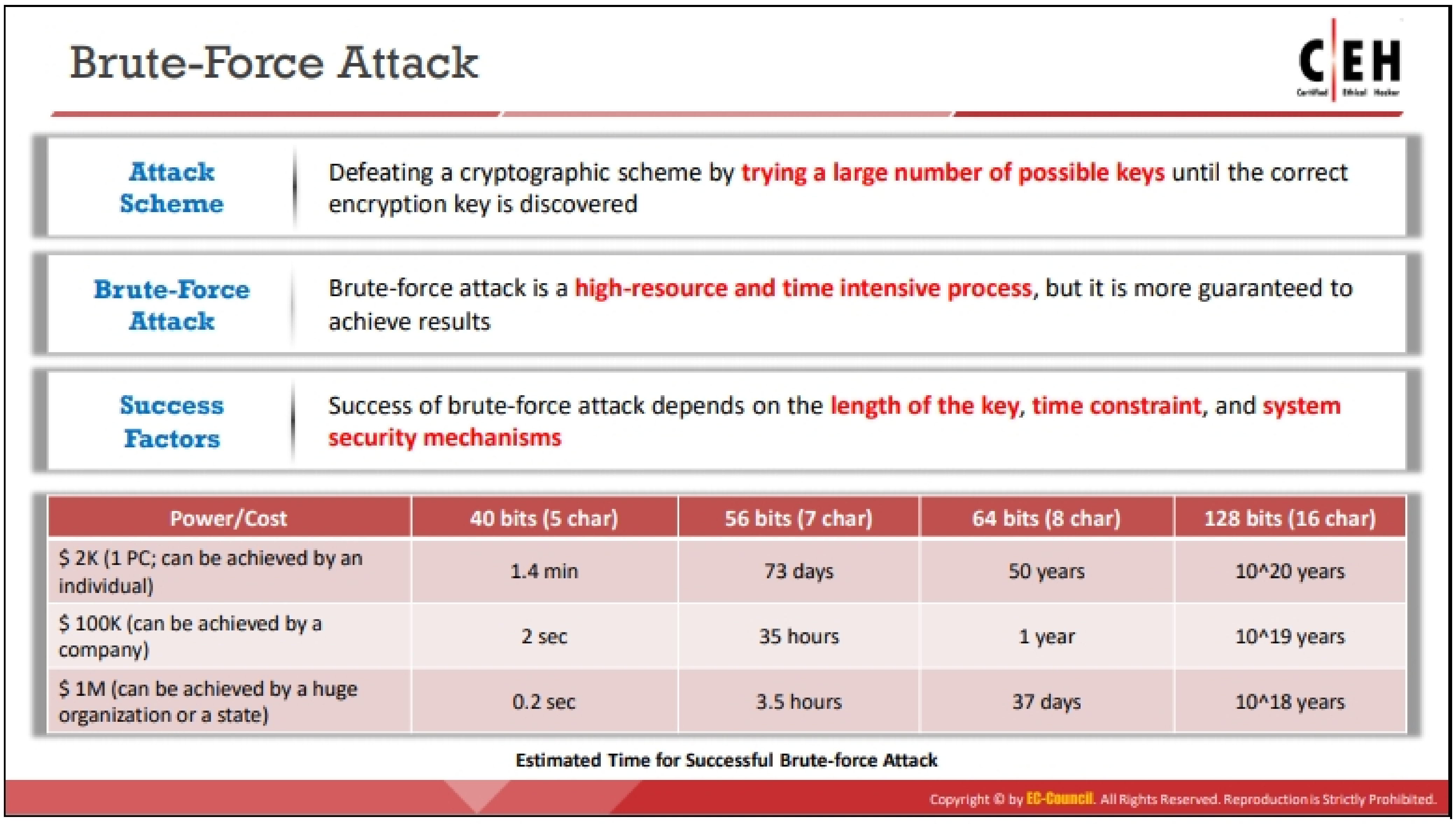
  In this type of attack, an attacker not only breaks a ciphertext but also breaks into a larger system, which is dependent of that ciphertext. The attacker usually breaks an n-bit key cipher into $2^{n/2}$ operations. Once an attacker breaks the cipher, he gets access to the system, and he can control the whole system, access confidential data, and perform further attacks.

- **Timing Attack**

  It is based on repeatedly measuring the exact execution times of modular exponentiation operations. The attacker tries to break the ciphertext by analyzing the time taken to execute the encryption and decryption algorithm for various inputs. In a computer, the time taken to execute a logical operation may vary based on the input given. An attacker tries to extract the plaintext by giving varying inputs.

- **Man-in-the-Middle Attack**

  This attack is performed against a cryptographic protocol. Here, an attacker intercepts the communication between a client and a server and negotiates the cryptographic parameters. Using this attack, an attacker can decrypt the encrypted content and obtain confidential information such as system passwords. An attacker can also inject commands that can modify the data in transit. The attacker usually performs an MITM attack on public-key cryptosystems where key exchange is required before communication takes place.



Brute-Force Attack

Attack Scheme	Defeating a cryptographic scheme by **trying a large number of possible keys** until the correct encryption key is discovered
Brute-Force Attack	Brute-force attack is a **high-resource and time intensive process**, but it is more guaranteed to achieve results
Success Factors	Success of brute-force attack depends on the **length of the key**, **time constraint**, and **system security mechanisms**

Power/Cost	40 bits (5 char)	56 bits (7 char)	64 bits (8 char)	128 bits (16 char)
$ 2K (1 PC; can be achieved by an individual)	1.4 min	73 days	50 years	10^20 years
$ 100K (can be achieved by a company)	2 sec	35 hours	1 year	10^19 years
$ 1M (can be achieved by a huge organization or a state)	0.2 sec	3.5 hours	37 days	10^18 years

Estimated Time for Successful Brute-force Attack

Brute-Force Attack

It is extremely difficult to crack cryptographic systems, as they have no practical weaknesses to exploit; however, it is not impossible. Cryptographic systems use cryptographic algorithms to encrypt a message. These cryptographic algorithms use a key to encrypt or decrypt messages. In cryptography, this key is the important parameter that specifies the transformation of plaintext to ciphertext and vice versa. If you are able to guess or find the key used for decryption, then you can decrypt the messages and read them in clear text. 128-bit keys are common and considered strong. From a security perspective, to avoid guessing the key, cryptographic systems use randomly generated keys. This makes you devote considerable effort toward guessing the key. However, you still have a choice to determine the key used for encryption or decryption.

You can attempt to decrypt a message using all possible keys until you discover the key used for encryption. This method of discovering a key is called a brute-force attack. However, doing so requires a massive amount of processing power. It is a resource-intensive and time-intensive process. For any non-flawed protocol, the average time needed to find the key in a brute-force attack depends on the length of the key. If the key length is short, then it will take less time to find the key; if it is long, it will take more time. A brute-force attack will be successful if and only if the attacker has enough time to discover the key. However, the time required is relative to the length of the key.

The difficulty of a brute-force attack depends on various factors, such as

- The length of the key
- The number of possible values each component of the key can have
- The time it takes to attempt each key

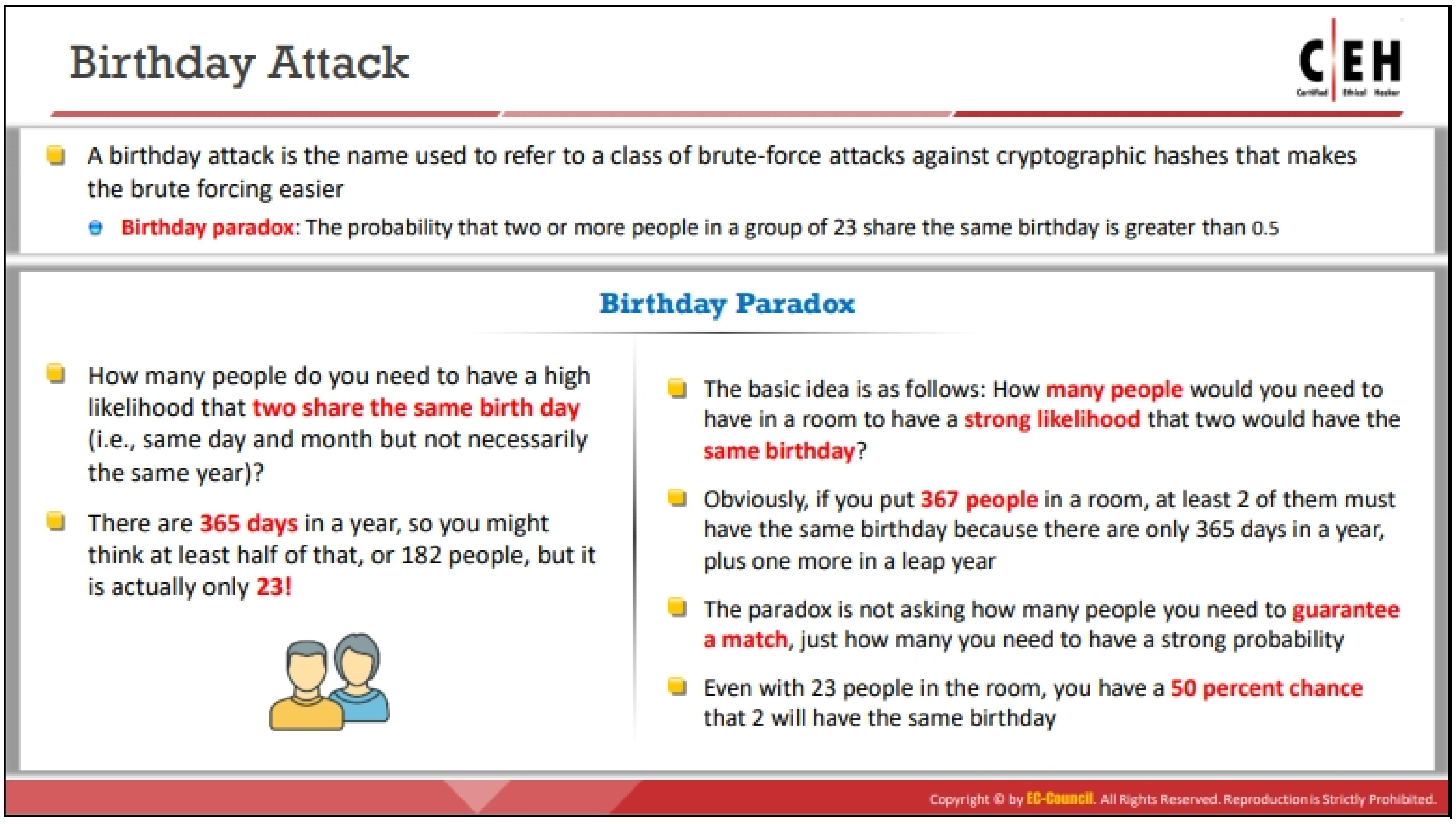
- If there is any mechanism that locks the attacker out after a certain number of failed attempts

For example, if a system could brute-force a DES 56-bit key in one second, then for an AES 128-bit key, it takes approximately 149 trillion years. To perform a brute-force attack, the attacker needs double the time for every additional bit of key length; the reason is that the number of keys doubles with an increase of one bit.

However, a brute-force attack is more likely to achieve results.

Power/Cost	**40 bits (5 char)**	**56 bits (7 char)**	**64 bits (8 char)**	**128 bits (16 char)**
$ 2K (1 PC. Can be achieved by an individual)	1.4 min	73 days	50 years	10^20 years
$ 100K (this can be achieved by a company)	2 sec	35 hours	1 year	10^19 years
$ 1M (Achieved by a huge organization or a state)	0.2 sec	3.5 hours	37 days	10^18 years

Table 20.5: Estimate time for a successful brute-force attack



Birthday Attack

- A birthday attack is the name used to refer to a class of brute-force attacks against cryptographic hashes that makes the brute forcing easier
  - **Birthday paradox**: The probability that two or more people in a group of 23 share the same birthday is greater than 0.5

Birthday Paradox

- How many people do you need to have a high likelihood that **two share the same birth day** (i.e., same day and month but not necessarily the same year)?
- There are **365 days** in a year, so you might think at least half of that, or 182 people, but it is actually only **23!**

- The basic idea is as follows: How **many people** would you need to have in a room to have a **strong likelihood** that two would have the **same birthday**?
- Obviously, if you put **367 people** in a room, at least 2 of them must have the same birthday because there are only 365 days in a year, plus one more in a leap year
- The paradox is not asking how many people you need to **guarantee a match**, just how many you need to have a strong probability
- Even with 23 people in the room, you have a **50 percent chance** that 2 will have the same birthday

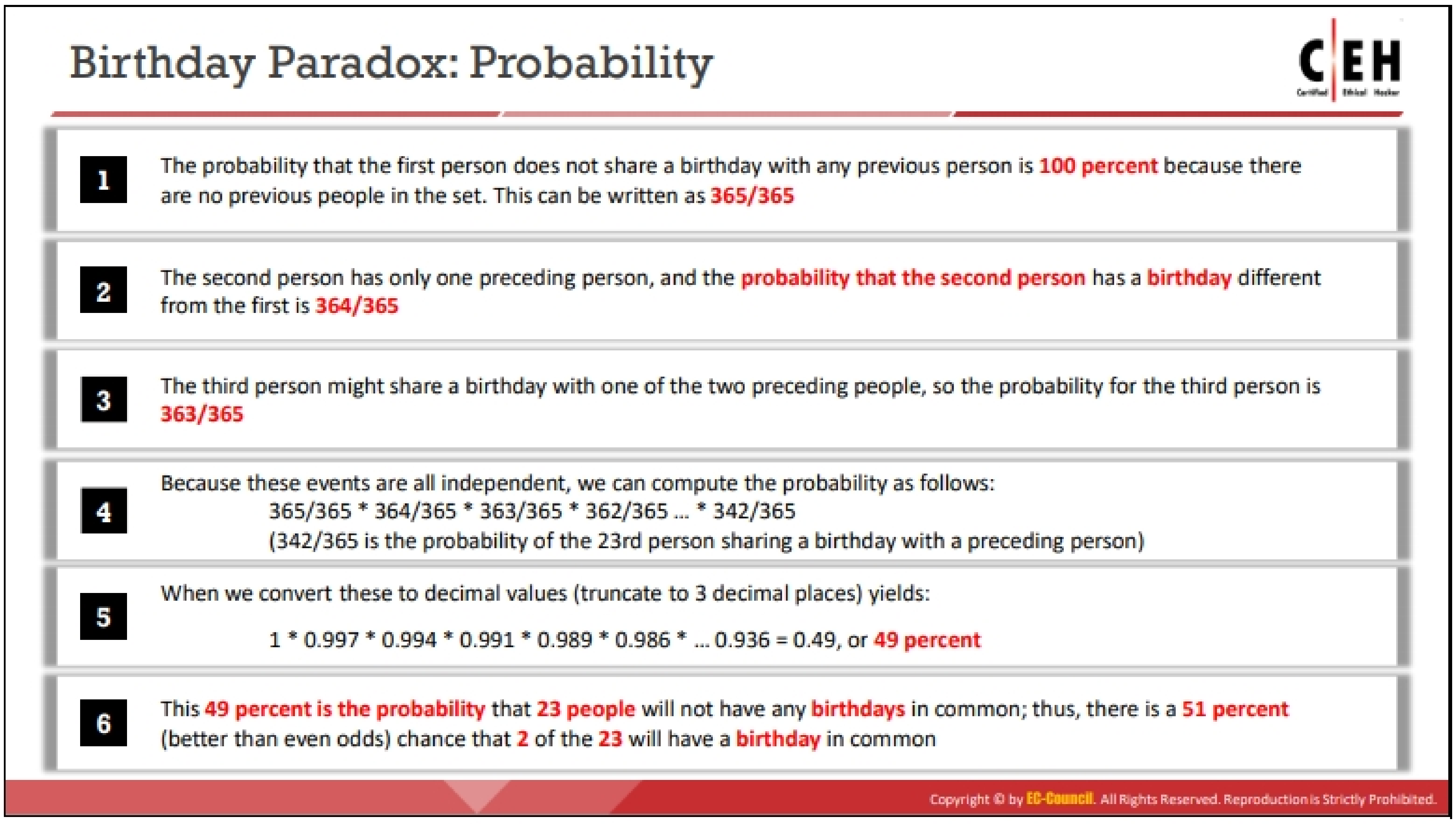
Birthday Attack

A birthday attack refers to a class of brute-force attacks against cryptographic hashes that renders brute-forcing easier to performs. This attack depends on the birthday paradox, which is the probability of two or more people in a group of 23 sharing the same birthday is greater than 0.5.

Birthday Paradox

For example, how many people are needed to have a high likelihood that two will share the same birthday (i.e., same day and month, not year). There are 365 days a year, and therefore, you might think that at least half or 182 people share the same birthday, when it is actually only 23!

The basic idea is as follows: How many people would you need to have in a room to have a strong likelihood that two amongst them would have the same birthday (same day and month, but not year). Obviously, if you put 367 people in a room, at least two of them must have their birthdays on the same day and month since there are only 365 days in a year, and an additional day in the case of a leap year. The paradox is not the number of people you need to guarantee a match, but the number of people you need to have a strong probability. Even with 23 people in a room, there is a 50% chance that two them will have their birthdays on the same day and month.



Birthday Paradox: Probability

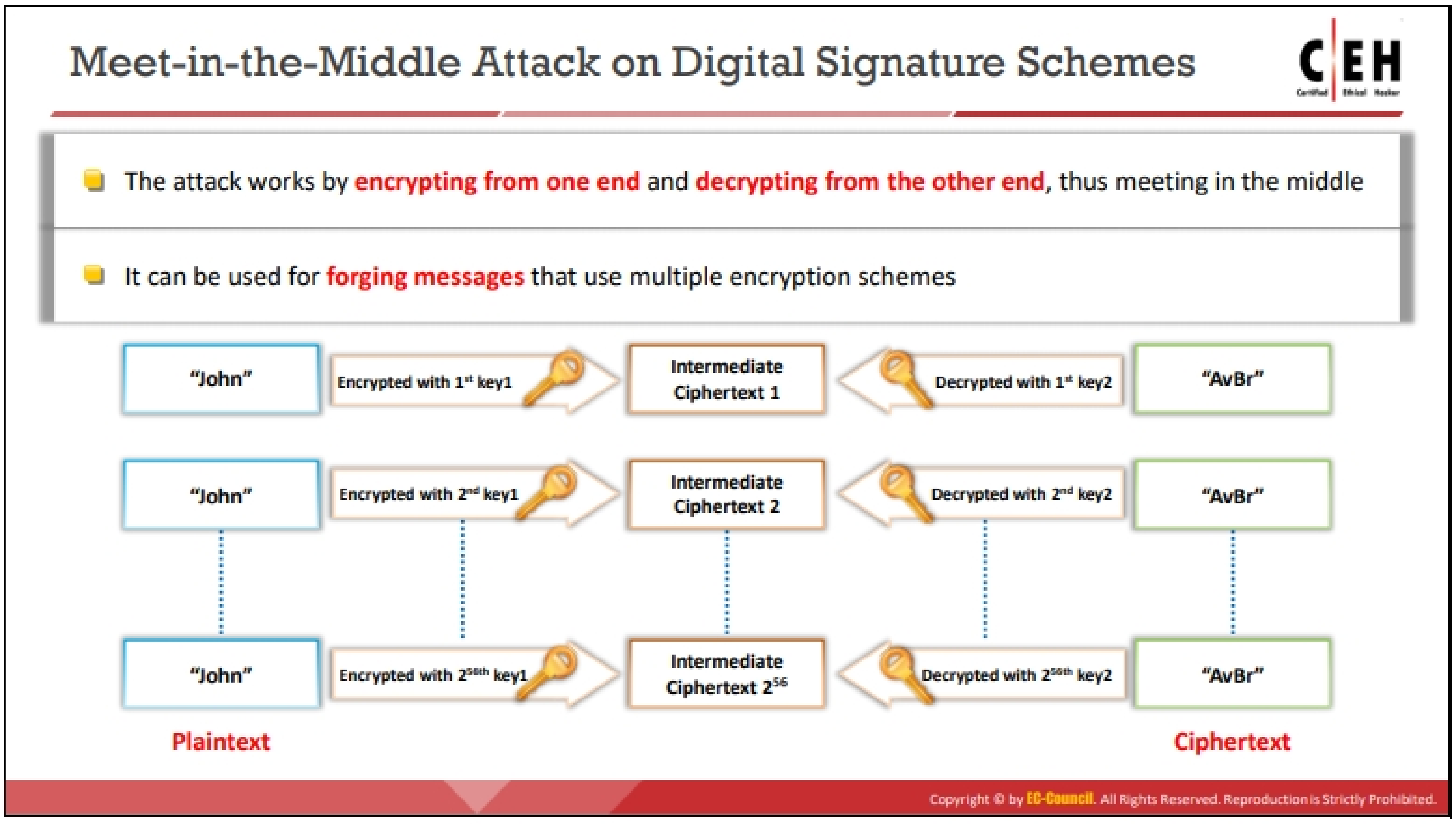
1. The probability that the first person does not share a birthday with any previous person is **100 percent** because there are no previous people in the set. This can be written as **365/365**
2. The second person has only one preceding person, and the **probability that the second person** has a **birthday** different from the first is **364/365**
3. The third person might share a birthday with one of the two preceding people, so the probability for the third person is **363/365**
4. Because these events are all independent, we can compute the probability as follows:
   365/365 * 364/365 * 363/365 * 362/365 ... * 342/365
   (342/365 is the probability of the 23rd person sharing a birthday with a preceding person)
5. When we convert these to decimal values (truncate to 3 decimal places) yields:
   1 * 0.997 * 0.994 * 0.991 * 0.989 * 0.986 * ... 0.936 = 0.49, or **49 percent**
6. This **49 percent is the probability** that **23 people** will not have any **birthdays** in common; thus, there is a **51 percent** (better than even odds) chance that **2** of the **23** will have a **birthday** in common

Birthday Paradox: Probability

The probability that the first person does not share a birthday with any previous person is 100% because there are no previous people in the set. This can be written as 365/365. The second person has only one preceding person, and the odds that the second person has a birthday different from the first are 364/365. The third person might share a birthday with two preceding people, so the odds of sharing a birthday with either of the two preceding people are 363/365. Because each of these are independent, we can compute the probability as follows: 365/365 * 364/365 * 363/365 * 362/365 ... * 342/365 (342 is the probability of the 23rd person who shares a birthday with a preceding person). When we convert these to decimal values, it yields (truncating at the third decimal point) 1 * 0.997 * 0.994 * 0.991 * 0.989 * 0.986 * ... 0.936 = 0.49 or 49%. This is the probability that 23 people will not have any birthdays in common; thus, there is a 51% (better than even odds) chance that two of the 23 will have a birthday in common.

The idea behind the birthday attack is to attempt to find a collision for a given hash. Now, assume that the hash is MD5 with a 128-bit output. You would have to try 2^128 possible hashes to guarantee a collision, which is a very large number. In decimal notation, it is 3.4028236692093846346337460743177e+38

Now, from the birthday paradox, we need 1.174√2^128 or 21656477542535013597.184 hashes to guarantee a collision. Furthermore, this is still a very large number but many orders of magnitude smaller than the abovementioned value.



Meet-in-the-Middle Attack on Digital Signature Schemes

- The attack works by **encrypting from one end** and **decrypting from the other end**, thus meeting in the middle
- It can be used for **forging messages** that use multiple encryption schemes

"John"
Encrypted with 1st key1
Intermediate Ciphertext 1
Decrypted with 1st key2
"AvBr"
"John"
Encrypted with 2nd key1
Intermediate Ciphertext 2
Decrypted with 2nd key2
"AvBr"
"John"
Encrypted with 2^{56}th key1
Intermediate Ciphertext 2^{56}
Decrypted with 2^{56}th key2
"AvBr"
Plaintext
Ciphertext

Meet-in-the-Middle Attack on Digital Signature Schemes

A meet-in-the-middle attack is the best attack method for cryptographic algorithms using multiple keys for encryption. This attack reduces the number of brute-force permutations required to decode text encrypted by more than one key. A meet-in-the-middle attack uses space-time trade-off; it is also a type of birthday attack because it exploits the mathematics behind the birthday paradox, and the attack consumes less time than an exhaustive attack. It is called a meet-in-the-middle attack because it works by encrypting from one end and decrypting from the other end, thereby meeting "in the middle."

In the meet-in-the-middle attack, the attacker uses a known plaintext message. The attacker has access to both the plaintext as well as the respective encrypted text. This attack is performed by attackers for forging messages that use multiple encryption schemes.

Consider an example where the plaintext is "**John**," and the resulting double-DES-encrypted message is "**AvBr.**" To recover both the keys (i.e., key1 and key2) used for encryption, the attacker performs a brute-force attack on key1 using all the **2^{56} different single DES** possible keys to encrypt the plaintext of "John" and saves each key and the resulting intermediate ciphertext in a table. The attacker brute-forces key2 and decrypts "AvBr" up to **2^{56} times**. The attack is successful when the second brute-force attack gives the same result as the intermediate ciphertext present in the ciphertext table after the first brute-force attack. Once the attacker finds a match, he/she can determine both keys and complete the attack. At most, this attack takes 2^{56} or a maximum of **2^{57} total operations**. This enables the attacker to gain access to the data easily compared with double DES.

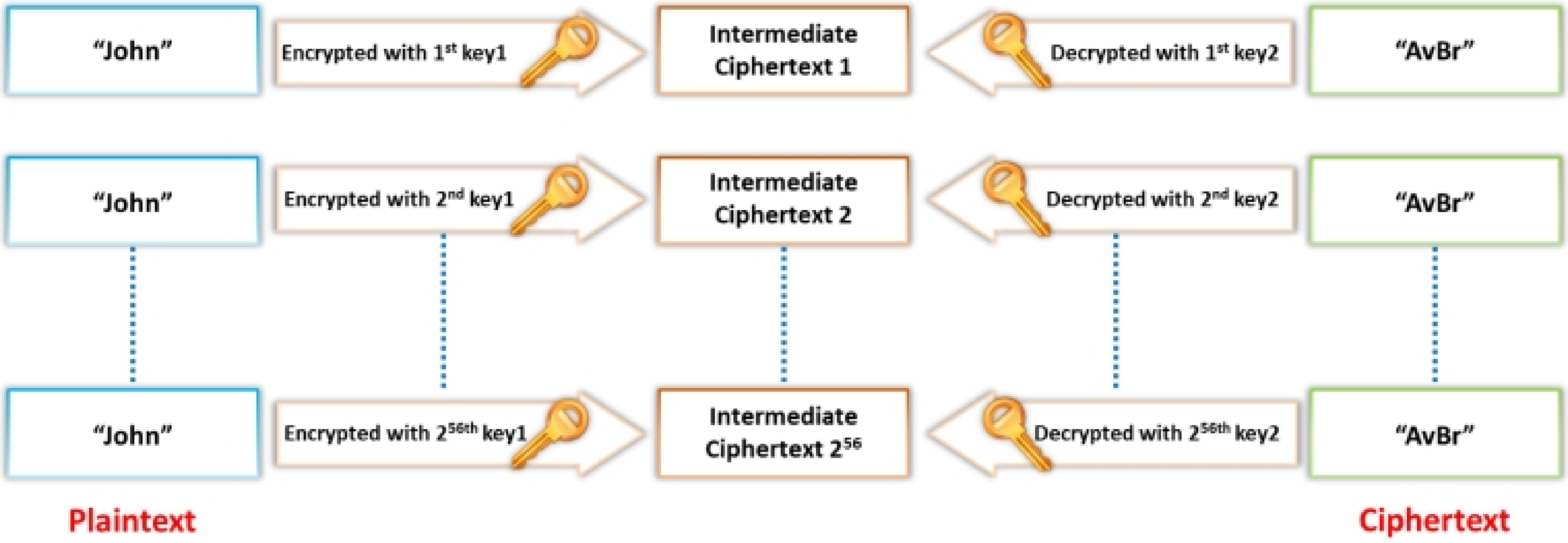
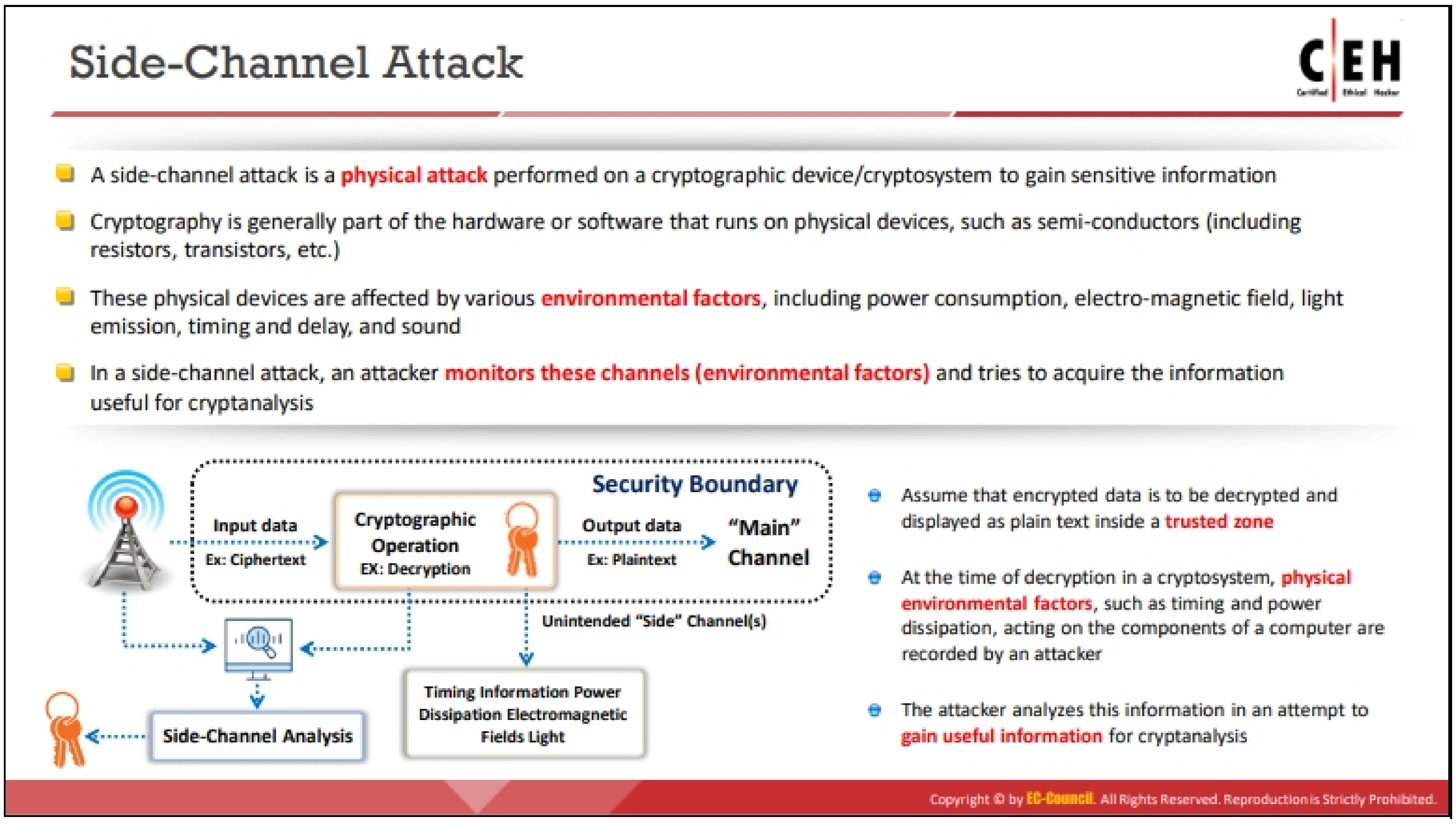

"John"
Encrypted with 1st key1
Intermediate Ciphertext 1
Decrypted with 1st key2
"AvBr"
"John"
Encrypted with 2nd key1
Intermediate Ciphertext 2
Decrypted with 2nd key2
"AvBr"
"John"
Encrypted with 256th key1
Intermediate Ciphertext 2^56
Decrypted with 256th key2
"AvBr"
Plaintext
Ciphertext

Figure 20.37: Meet-in-the-Middle attack on digital signature schemes



Side-Channel Attack

- A side-channel attack is a **physical attack** performed on a cryptographic device/cryptosystem to gain sensitive information
- Cryptography is generally part of the hardware or software that runs on physical devices, such as semi-conductors (including resistors, transistors, etc.)
- These physical devices are affected by various **environmental factors**, including power consumption, electro-magnetic field, light emission, timing and delay, and sound
- In a side-channel attack, an attacker **monitors these channels (environmental factors)** and tries to acquire the information useful for cryptanalysis

- Assume that encrypted data is to be decrypted and displayed as plain text inside a **trusted zone**
- At the time of decryption in a cryptosystem, **physical environmental factors**, such as timing and power dissipation, acting on the components of a computer are recorded by an attacker
- The attacker analyzes this information in an attempt to **gain useful information** for cryptanalysis

Side-Channel Attack

A side-channel attack is a physical attack performed on a cryptographic device/cryptosystem to gain sensitive information. Cryptography is generally part of the hardware or software that runs on physical devices such as semi-conductors (resistor, transistor, and so on) that interact with and affect various environmental factors as follows:

- **Power Consumption**

  Reveals operations that take place and parameters involved. It is applicable only to hardware cryptosystems. Power consumption analysis is of two types:

  - **Simple Power Analysis (SPA)**: Provides information regarding the instruction being executed at a certain time and the values of input and output
  - **Differential Power Analysis (DPA)**: It does not require the knowledge of the details of algorithm implementation; it exploits statistical methods

- **Electromagnetic Field**

  Computer components often generate electromagnetic radiation. By measuring the variations of the electromagnetic field over the chip surface, an attacker can predict its correlation to the underlying computation and data and may be able to deduce some valuable information about this computation and data.

- **Light Emission**

  Kuhn found that the average luminosity of a cathode ray tube (CRT) diffuse reflection of a wall is sufficient to reconstruct the signal displayed on the CRT. Thus, an attacker can gather ample information by reading the signals that a trusted computing platform's optical output channels emit.

According to Loughry and Umphress, one can deduce the data a computer is processing based on the optical radiation emitted from its LED (light-emitting diode) status indicators.

- **Timing and Delay**

  Systems often compute cryptographic algorithms without time consistency owing to performance optimizations. If such computations involves secret data, then the variations in time can be used to infer the secret information. Here, the attacker analyzes the time taken by a cryptographic device to process each message to discover the secret parameters.

- **Sound**

  Acoustic attacks exploit the sound produced during a computation. These acoustic emissions are from keyboards and computing components (e.g., CPU, memory)

In a side-channel attack, an attacker monitors these channels (environmental factors) and tries to acquire useful information for cryptanalysis. The information thus acquired is termed as side-channel information. Side-channel attacks are different from traditional/theoretical forms of attacks such as brute-force attacks. The side-channel attack depends on the way in which systems implement cryptographic algorithms rather than the algorithm itself.

Mitigation techniques for side-channel-attacks include the following:

- Use differential power analysis (DPA) proof protocols with delimited side-channel leakage characteristics and update the keys before the leakage accumulation is significant
- Use fixed-time algorithms (i.e., no data-dependent delays)
- Mask and blind algorithms using random nonces
- Implement differential matching techniques to minimize net data-dependent leakage from logic-level transitions
- Pre-charge registers and busses to remove leakage signatures from predictable data transitions
- Add amplitude or temporal noise to reduce the attacker's signal-to-noise ratio

Side-Channel Attack – Scenario

Assume that encrypted data are to be decrypted and displayed as plaintext inside a trusted zone. At the time of decryption in a cryptosystem, physical environmental factors, such as timing and power dissipation, acting on the components of a computer are recorded by an attacker. The attacker then analyzes this information to gain useful information for cryptanalysis.

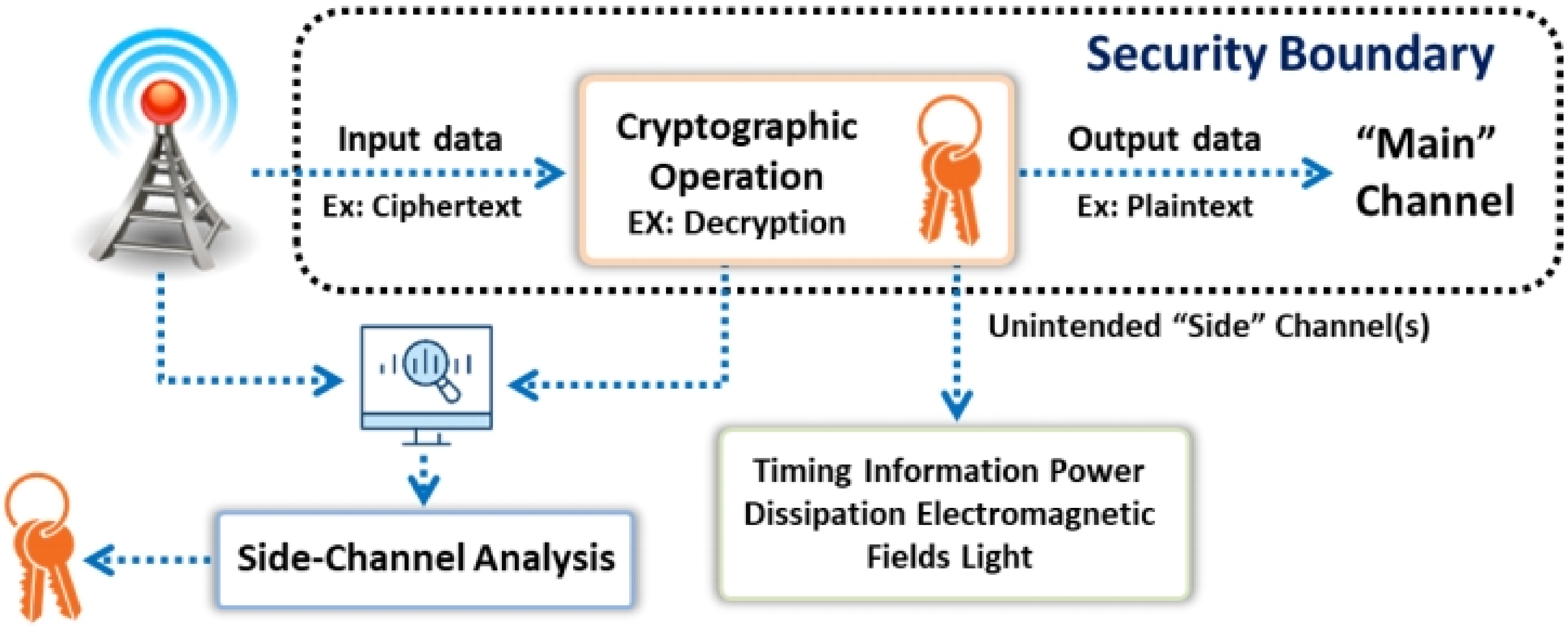
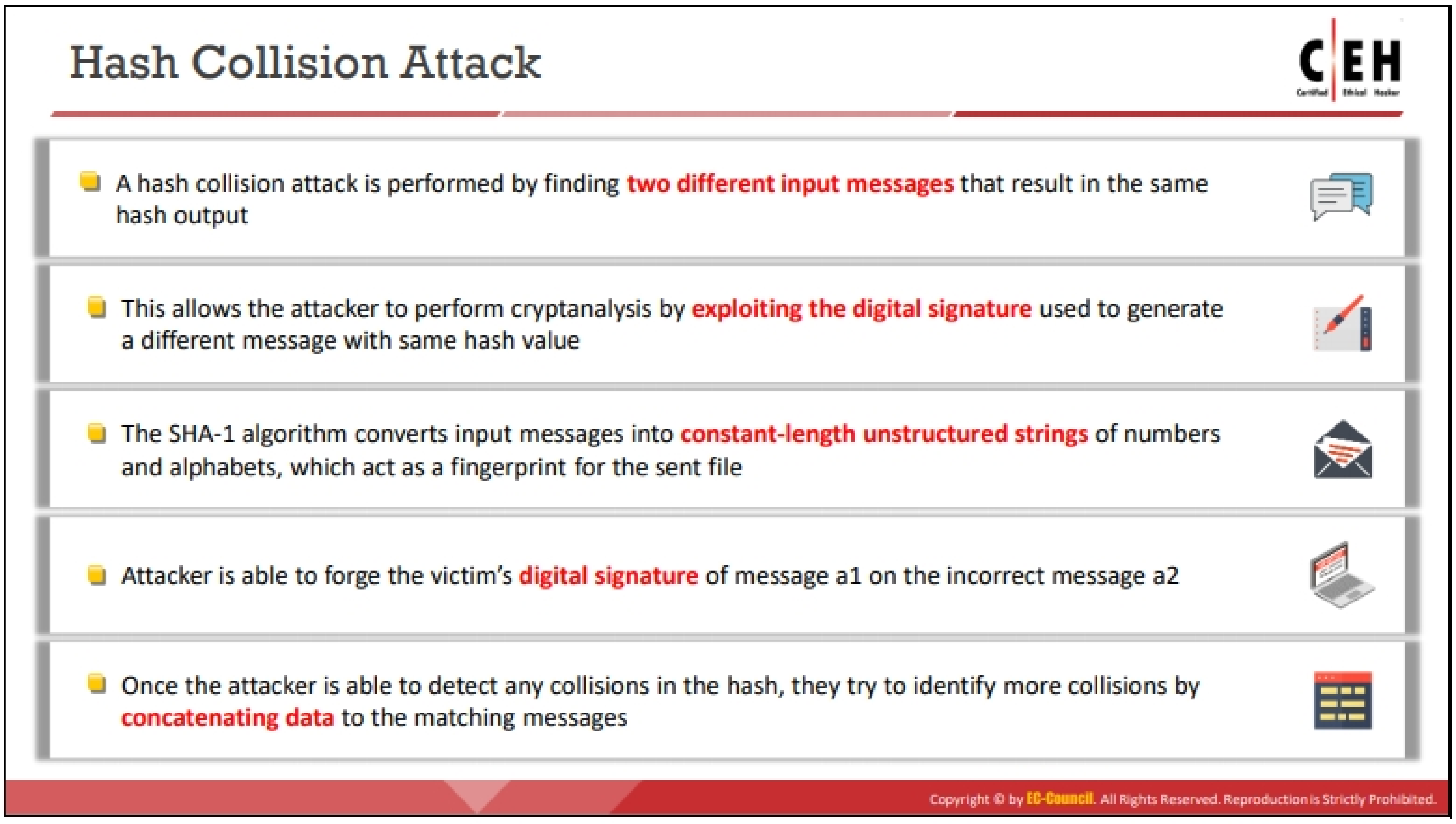

Security Boundary
Input data
Ex: Ciphertext
Cryptographic Operation
EX: Decryption
Output data
Ex: Plaintext
"Main" Channel
Unintended "Side" Channel(s)
Timing Information Power Dissipation Electromagnetic Fields Light
Side-Channel Analysis

Figure 20.38: Side-Channel attack – scenario



Hash Collision Attack

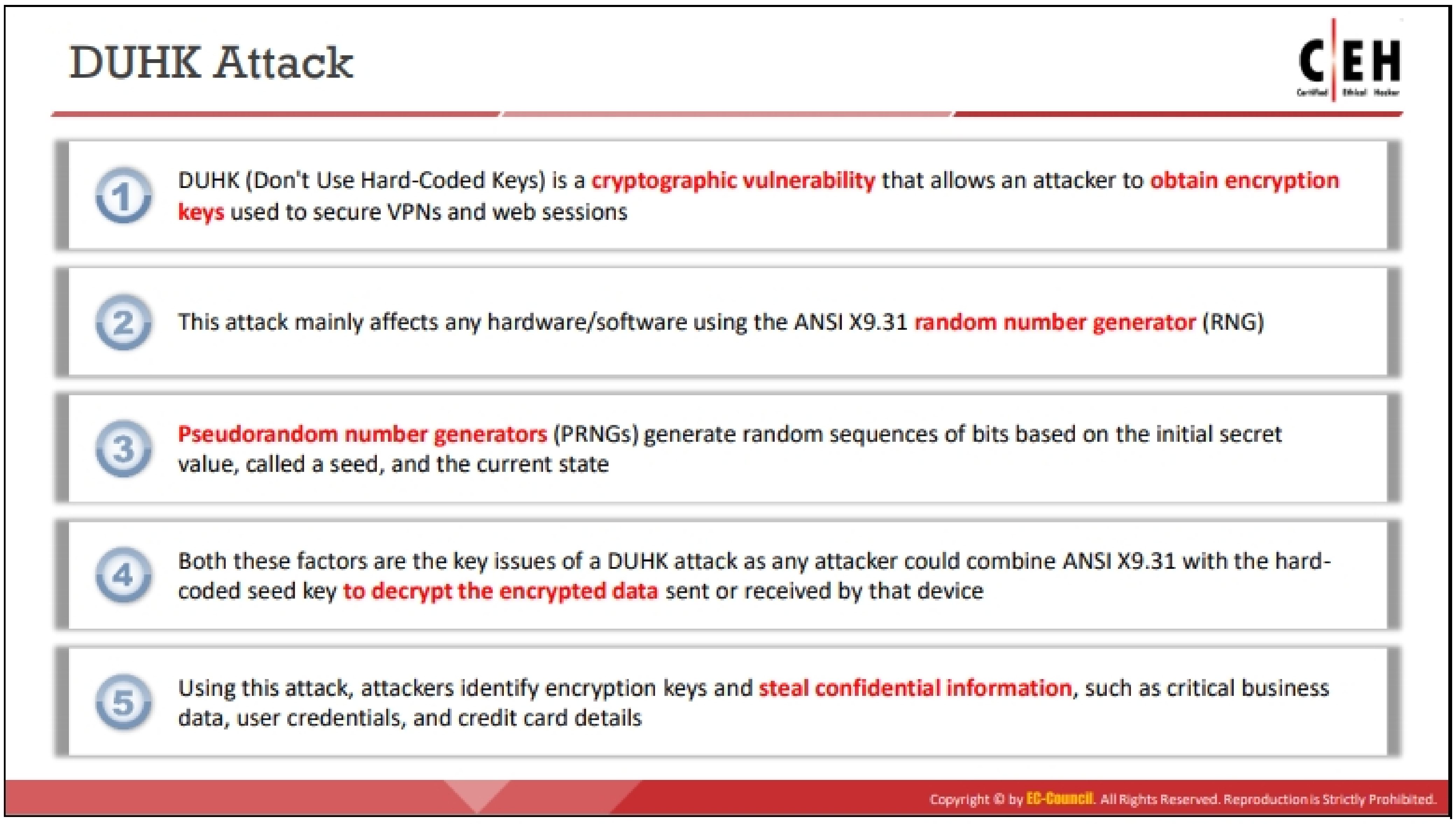
- A hash collision attack is performed by finding **two different input messages** that result in the same hash output
- This allows the attacker to perform cryptanalysis by **exploiting the digital signature** used to generate a different message with same hash value
- The SHA-1 algorithm converts input messages into **constant-length unstructured strings** of numbers and alphabets, which act as a fingerprint for the sent file
- Attacker is able to forge the victim's **digital signature** of message a1 on the incorrect message a2
- Once the attacker is able to detect any collisions in the hash, they try to identify more collisions by **concatenating data** to the matching messages

Hash Collision Attack

A hash collision attack is performed by finding two different input messages that result in the same hash output. For example, in a hash collision attack, "**hash(a1) = hash(a2)",** where a1 and a2 represent some random messages. Since the algorithm itself randomly selects these messages, attackers have no role in the content of these messages. This allows the attacker to perform cryptanalysis by exploiting the digital signature used to generate a different message with the same hash value.

One of the most popular hash functions is SHA-1, which is widely used as a digital signature algorithm. SHA-1 converts an input message into a constant length of unstructured strings of numbers and alphabets, which act as a fingerprint for the sent file. Therefore, the attacker tries to identify similar hashed output to get the digital signatures of the victim. This allows the attacker to forge the victim's digital signature of message a1 on message a2.

Once the attacker detects a collision in the hash, he/she can identify more collisions by concatenating the data to matching messages.



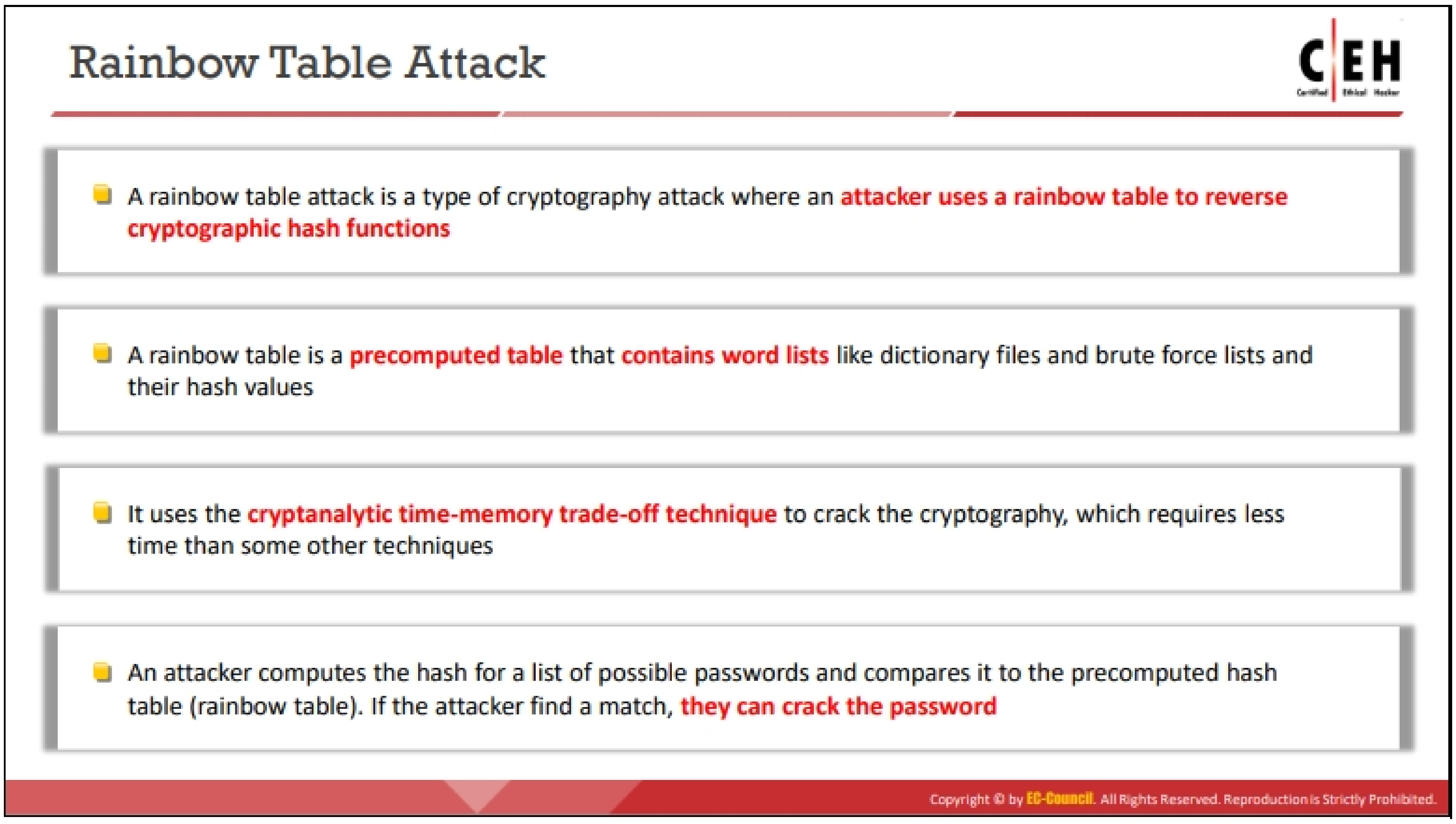
DUHK Attack

1. DUHK (Don't Use Hard-Coded Keys) is a **cryptographic vulnerability** that allows an attacker to **obtain encryption keys** used to secure VPNs and web sessions
2. This attack mainly affects any hardware/software using the ANSI X9.31 **random number generator** (RNG)
3. **Pseudorandom number generators** (PRNGs) generate random sequences of bits based on the initial secret value, called a seed, and the current state
4. Both these factors are the key issues of a DUHK attack as any attacker could combine ANSI X9.31 with the hard-coded seed key **to decrypt the encrypted data** sent or received by that device
5. Using this attack, attackers identify encryption keys and **steal confidential information**, such as critical business data, user credentials, and credit card details

DUHK Attack

Don't Use Hard-Coded Keys (DUHK) is a cryptographic vulnerability that allows attackers to obtain encryption keys used to secure VPNs and web sessions. This attack mainly affects any hardware/software using the ANSI X9.31 Random Number Generator (RNG). Pseudorandom number generators (PRNGs) generate random sequences of bits based on the initial secret value, called seed, and the current state. The PRNG algorithm generates cryptographic keys that are used to establish a secure communication channel over the VPN. In some cases, the seed key is hardcoded into the implementation. Both the factors are key issues of the DUHK attack, as any attacker can combine ANSI X9.31 with the hard-coded seed key to decrypt the encrypted data sent or received by that device.

Man-in-the-middle attackers use the DUHK attack to learn the seed value, observe the current session, and obtain the current state value. Using this attack, attackers can identify encryption keys and steal confidential information such as critical business data, user credentials, and credit card details.



Rainbow Table Attack

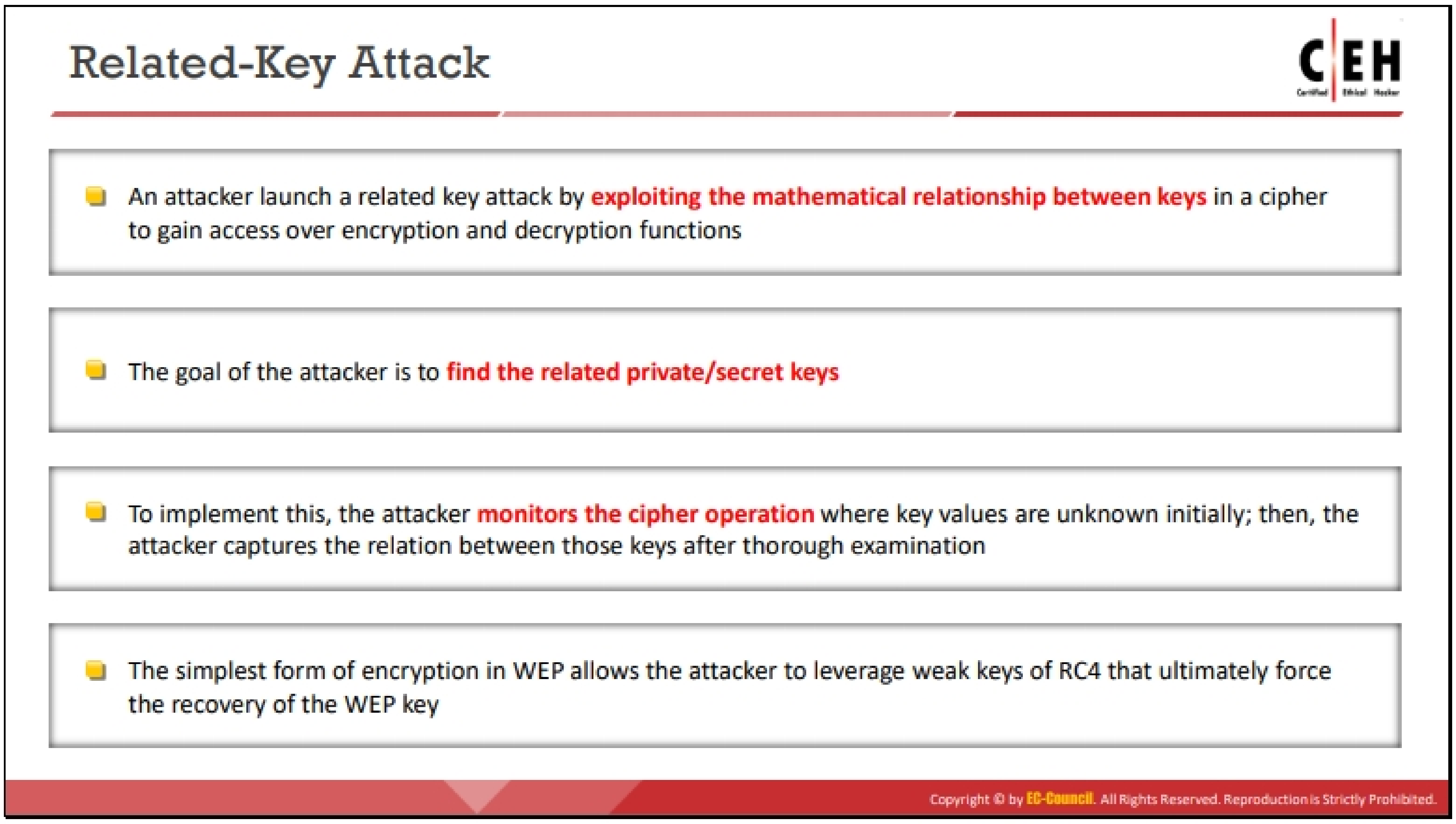
- A rainbow table attack is a type of cryptography attack where an **attacker uses a rainbow table to reverse cryptographic hash functions**
- A rainbow table is a **precomputed table** that **contains word lists** like dictionary files and brute force lists and their hash values
- It uses the **cryptanalytic time-memory trade-off technique** to crack the cryptography, which requires less time than some other techniques
- An attacker computes the hash for a list of possible passwords and compares it to the precomputed hash table (rainbow table). If the attacker find a match, **they can crack the password**

Rainbow Table Attack

A rainbow table attack is a type of cryptography attack whereby an attacker uses a rainbow table for reversing cryptographic hash functions. A rainbow table attack uses the cryptanalytic time-memory trade-off technique, which is less time consuming than other techniques. It uses already calculated information stored in memory for encryption. In the rainbow table attack, the attacker creates a table of all the possible passwords and their respective hash values, called a rainbow table, in advance.

A rainbow table contains word lists such as dictionary files and brute-force lists and their hash values. It is a lookup table particularly used for recovering a plaintext password from a ciphertext. The attacker uses this table to look for the password and tries to recover it from password hashes.

An attacker computes the hash for a list of possible passwords and compares it with the pre-computed hash table (rainbow table). If a match is found, then he/she can crack the password. It is easy to recover passwords by comparing the captured password hashes with pre-computed tables.



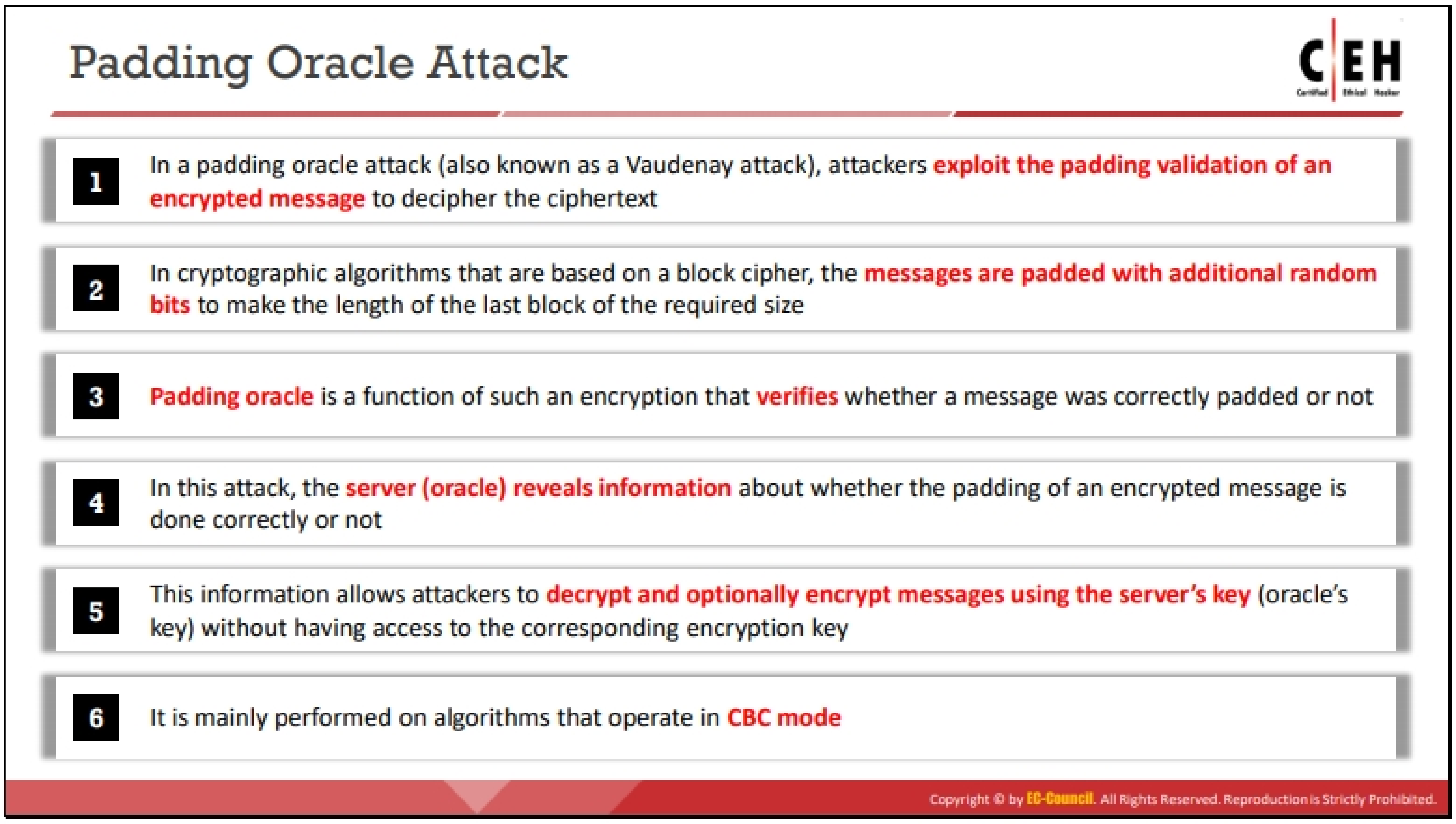
Related-Key Attack

- An attacker launch a related key attack by **exploiting the mathematical relationship between keys** in a cipher to gain access over encryption and decryption functions
- The goal of the attacker is to **find the related private/secret keys**
- To implement this, the attacker **monitors the cipher operation** where key values are unknown initially; then, the attacker captures the relation between those keys after thorough examination
- The simplest form of encryption in WEP allows the attacker to leverage weak keys of RC4 that ultimately force the recovery of the WEP key

Related-Key Attack

Attackers launch a related-key attack by exploiting the mathematical relationship between the keys in a cipher and gain access to encryption and decryption functions. The attacker's motive for launching this attack is to find the related private/secret keys. To implement this attack, the attacker monitors the cipher operation where key values are initially unknown; then, the attacker captures the relation between those keys after a thorough examination. For instance, the attacker observes and finds that the last 80 bits of the keys are same at all times, but he/she does not initially know about these bits.

The failure in the WEP cryptogram, i.e., when used in wireless networks, is the best example of this attack. In this attack, each AP and user interface device uses the same key. The encryption used in WEP is a stream cipher known as RC4; it is important to note that the same keys should not be repeated in the stream cipher. To avoid this, WEP integrates a 24-bit initial vector (IV) in every packet transferred. The RC4 key for that particular packet is the IV associated with the WEP key. WEP keys need to be changed manually, however, this is rarely done. Hence, the attacker notices that the keys used for encryption are often the same. This drawback poses various risks on WEP, especially using the birthday paradox, because for every 4096 packets, two parties will share the same IV and hence the same RC4 key. This simple form of encryption allows the attacker to leverage weak RC4 keys, which ultimately forces the recovery of the WEP key.



Padding Oracle Attack

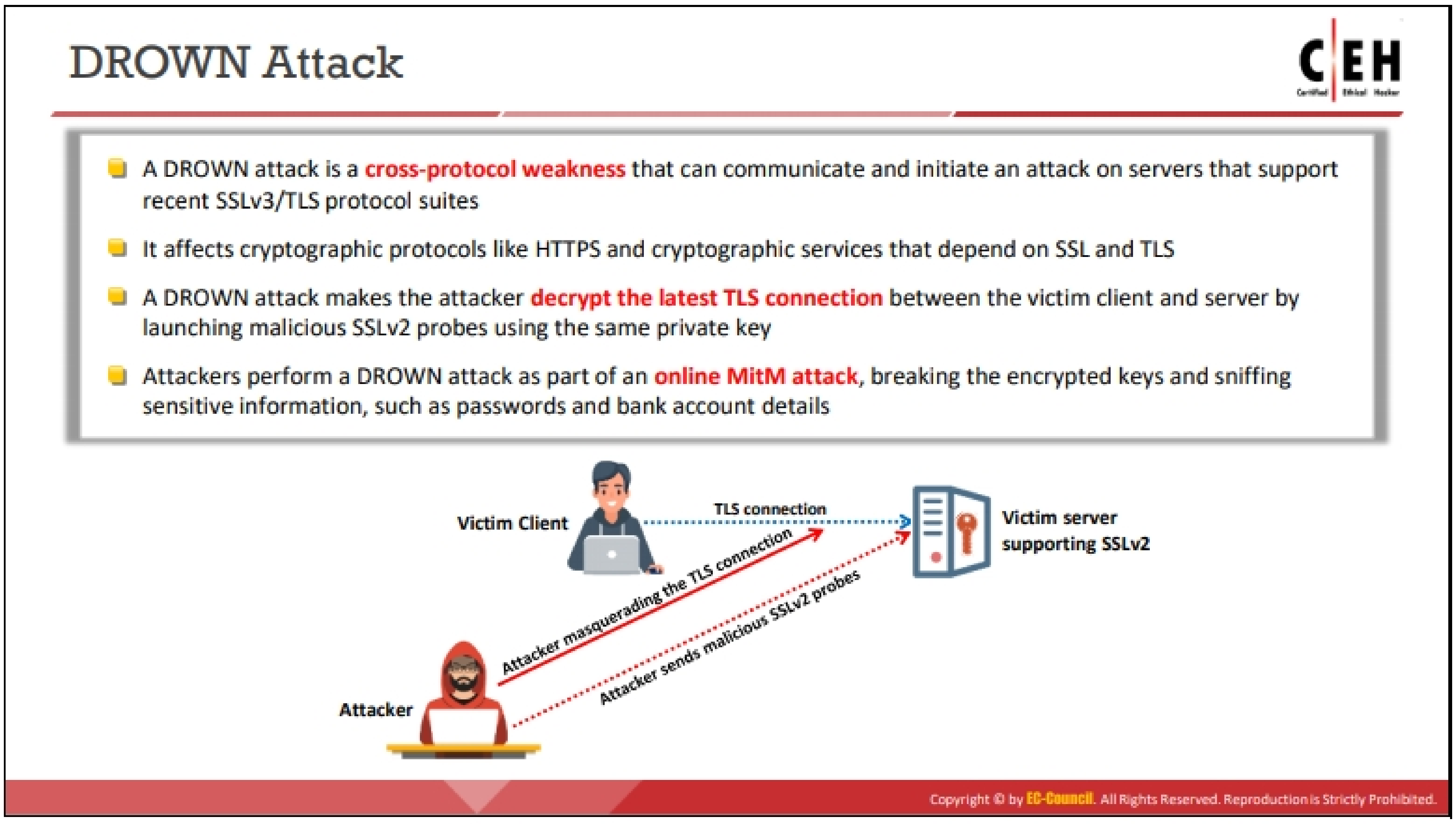
1. In a padding oracle attack (also known as a Vaudenay attack), attackers **exploit the padding validation of an encrypted message** to decipher the ciphertext
2. In cryptographic algorithms that are based on a block cipher, the **messages are padded with additional random bits** to make the length of the last block of the required size
3. **Padding oracle** is a function of such an encryption that **verifies** whether a message was correctly padded or not
4. In this attack, the **server (oracle) reveals information** about whether the padding of an encrypted message is done correctly or not
5. This information allows attackers to **decrypt and optionally encrypt messages using the server's key** (oracle's key) without having access to the corresponding encryption key
6. It is mainly performed on algorithms that operate in **CBC mode**

Padding Oracle Attack

In a padding oracle attack, the attackers exploit the padding validation of an encrypted message to decipher the ciphertext. Such an attack is also known as a Vaudenay attack. In many cryptographic algorithms based on a block cipher, the messages are padded with additional random bits so that the length of the last block is of the required size. Padding oracle is a function of such encryption that verifies if a message was correctly padded. This attack is mainly performed on algorithms that operate in the CBC (Cipher Block Chaining) mode.

In this attack, the server (oracle) reveals information about whether the padding of an encrypted message was correctly done. In some cases, this information allows attackers to decrypt and optionally encrypt messages using the server's key (oracle's key) without having access to the corresponding encryption key.

For example, consider a scenario with a standard implementation of CBC decryption. The server decrypts all the ciphertext blocks, verifies the padding, removes all the additional padding done during the encryption process, and then returns the original message to application or user. If, for example, the server is unable to decrypt the message due to a padding error and returns an error message "Decryption failure: Invalid Padding" instead of a generic message "Decryption failed", this information can be exploited by an attacker. The attacker can use the server as a padding oracle to decipher the encrypted messages.



DROWN Attack

- A DROWN attack is a **cross-protocol weakness** that can communicate and initiate an attack on servers that support recent SSLv3/TLS protocol suites
- It affects cryptographic protocols like HTTPS and cryptographic services that depend on SSL and TLS
- A DROWN attack makes the attacker **decrypt the latest TLS connection** between the victim client and server by launching malicious SSLv2 probes using the same private key
- Attackers perform a DROWN attack as part of an **online MitM attack**, breaking the encrypted keys and sniffing sensitive information, such as passwords and bank account details

Victim Client
TLS connection
Victim server supporting SSLv2
Attacker masquerading the TLS connection
Attacker sends malicious SSLv2 probes
Attacker

DROWN Attack

Decrypting RSA with Obsolete and Weakened eNcryption (DROWN) is a grave vulnerability that can affect important cryptographic protocols such as HTTPS and other cryptographic services that depend on SSL and TSL. The DROWN attack is a cross-protocol weakness that can communicate and initiate an attack on servers supporting recent SSLv3/TLS protocol suites. It is a new form of cross-protocol Bleichenbacher padding oracle attack.

The server is critically vulnerable to the DROWN attack if

- The server permits SSLv2 connection, which is mostly caused by a misconfiguration or incorrect default settings.
- The same private key certificate is used on a different server that allows SSLv2 connection, and it also makes the TLS server vulnerable, as the SSLv2 server can leak the key information.

The DROWN attack allows the attacker to decrypt the latest TLS connection between the victim client and the server by launching malicious SSLv2 probes using the same private key. Using this attack, the attacker can also force the victim client and server to use the RSA key exchange. Thus, the attacker can disrupt connections among the latest browsers and servers that favor the use of latest techniques, i.e., perfect-forward-secret key exchange, such as DHE and ECDH.

Attackers perform the DROWN attack as part of an online man-in-the-middle (MITM) attack, breaking encrypted keys, sniffing or stealing sensitive information such as passwords and bank account details, and accessing personal emails or messages. By performing this attack, the attacker can also masquerade as a secure website and thus seize or change the website contents.

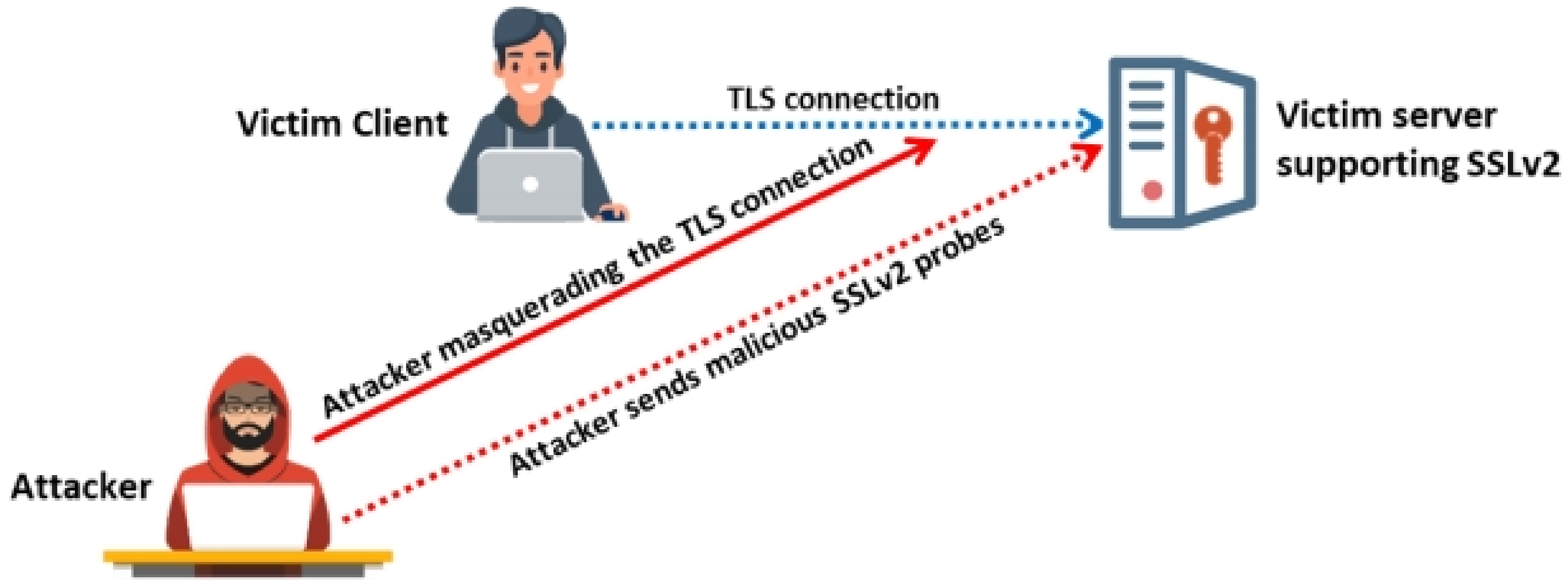
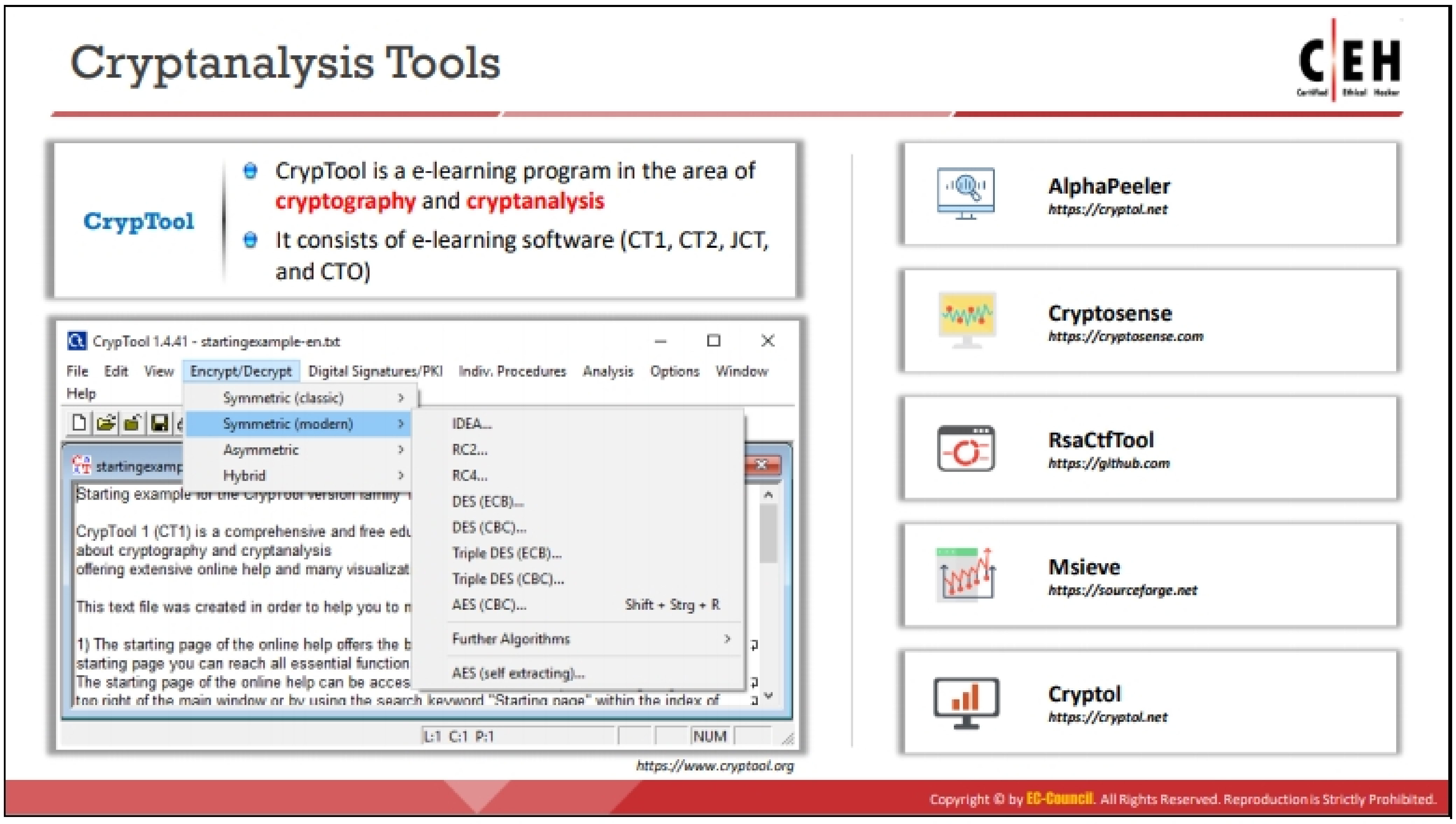

Victim Client
TLS connection
Victim server supporting SSLv2
Attacker masquerading the TLS connection
Attacker sends malicious SSLv2 probes
Attacker

Figure 20.39: DROWN attack



Cryptanalysis Tools

CrypTool

- CrypTool is a e-learning program in the area of **cryptography** and **cryptanalysis**
- It consists of e-learning software (CT1, CT2, JCT, and CTO)

CrypTool 1.4.41 - startingexample-en.txt
File
Edit
View
Encrypt/Decrypt
Digital Signatures/PKI
Indiv. Procedures
Analysis
Options
Window
Help
Symmetric (classic)
Symmetric (modern)
Asymmetric
Hybrid
IDEA...
RC2...
RC4...
DES (ECB)...
DES (CBC)...
Triple DES (ECB)...
Triple DES (CBC)...
AES (CBC)...
Shift + Strg + R
Further Algorithms
AES (self extracting)...
startingexamp
L:1 C:1 P:1
NUM

Starting example for the CrypTool version family

CrypTool 1 (CT1) is a comprehensive and free edu
about cryptography and cryptanalysis
offering extensive online help and many visualizat

This text file was created in order to help you to m

1) The starting page of the online help offers the b
starting page you can reach all essential function
The starting page of the online help can be acces
top right of the main window or by using the search keyword "Starting page" within the index of

https://www.cryptool.org

AlphaPeeler
https://cryptol.net

Cryptosense
https://cryptosense.com

RsaCtfTool
https://github.com

Msieve
https://sourceforge.net

Cryptol
https://cryptol.net

Cryptanalysis Tools

Attackers use cryptanalysis tools to analyze and break ciphers. Some cryptanalysis tools are discussed as follows.

- **CrypTool**

  Source: *https://www.cryptool.org*

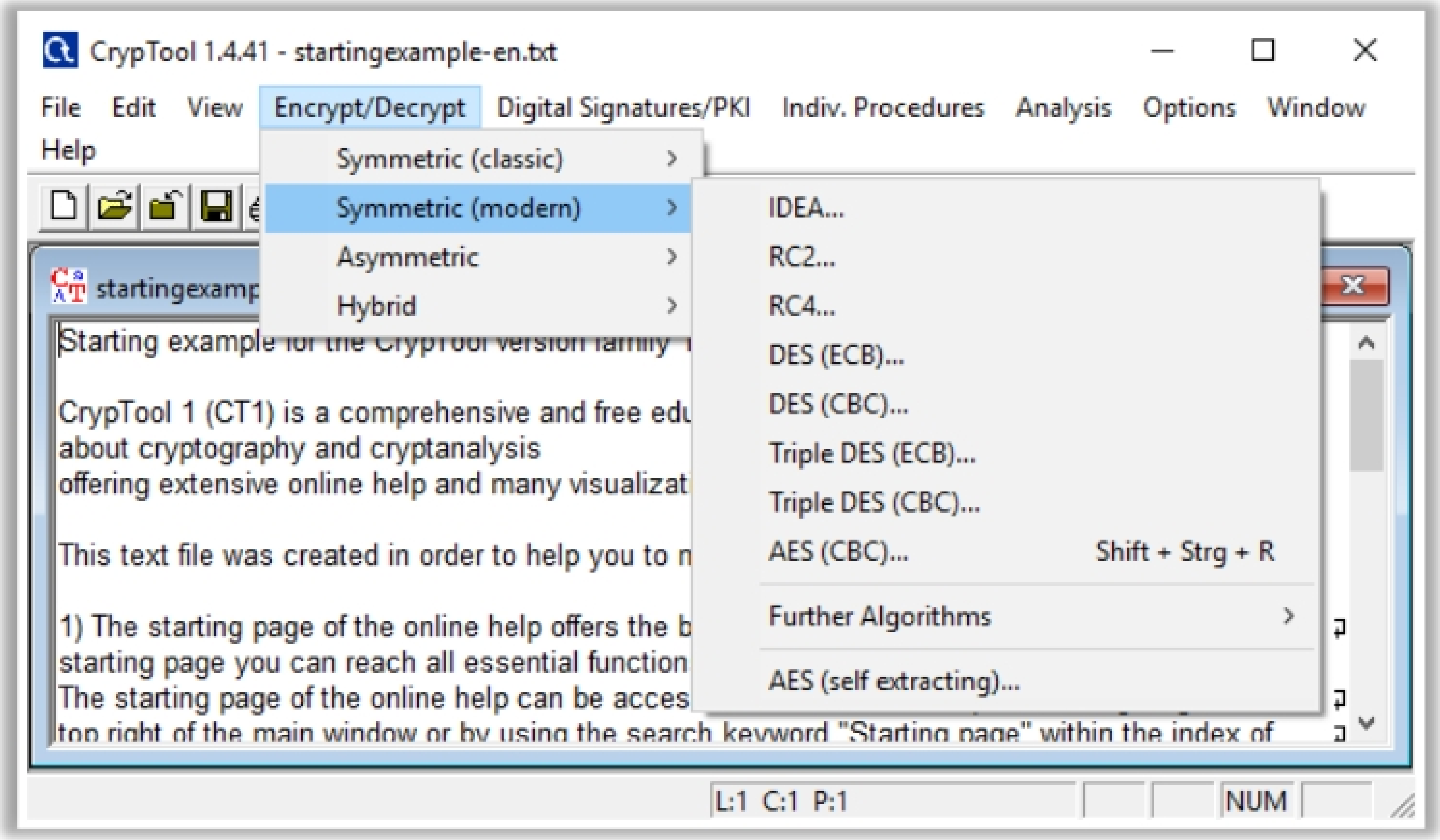
  The CrypTool project develops e-learning programs in the area of cryptography and cryptanalysis. It consists of e-learning software (CT1, CT2, JCT, and CTO).

  CrypTool 1 (CT1) – It is written in C++ and is a Windows program. It supports classical and modern cryptographic algorithms (encryption and decryption, key generation, secure passwords, authentication, secure protocols, etc.). It is used to perform cryptanalysis of several algorithms (Vigenère, RSA, AES, etc.)

  CrypTool 2 (CT2) – It supports visual programming GUI and execution of cascades of cryptographic procedures. It runs under Windows**.**

  JCrypTool (JCT) – It allows comprehensive cryptographic experimentation on Linux, MAC OS X, and Windows. It also allows users to develop and extend its platform in various ways with their own crypto plug-ins.

  CrypTool-Online (CTO) – It runs in a browser and provides a variety of encryption methods and analysis tools.


CrypTool 1.4.41 - startingexample-en.txt
File
Edit
View
Encrypt/Decrypt
Digital Signatures/PKI
Indiv. Procedures
Analysis
Options
Window
Help
Symmetric (classic)
Symmetric (modern)
Asymmetric
Hybrid
IDEA...
RC2...
RC4...
DES (ECB)...
DES (CBC)...
Triple DES (ECB)...
Triple DES (CBC)...
AES (CBC)... Shift + Strg + R
Further Algorithms
AES (self extracting)...
startingexamp
L:1 C:1 P:1
NUM

Starting example

CrypTool 1 (CT1) is a comprehensive and free edu
about cryptography and cryptanalysis
offering extensive online help and many visualizat

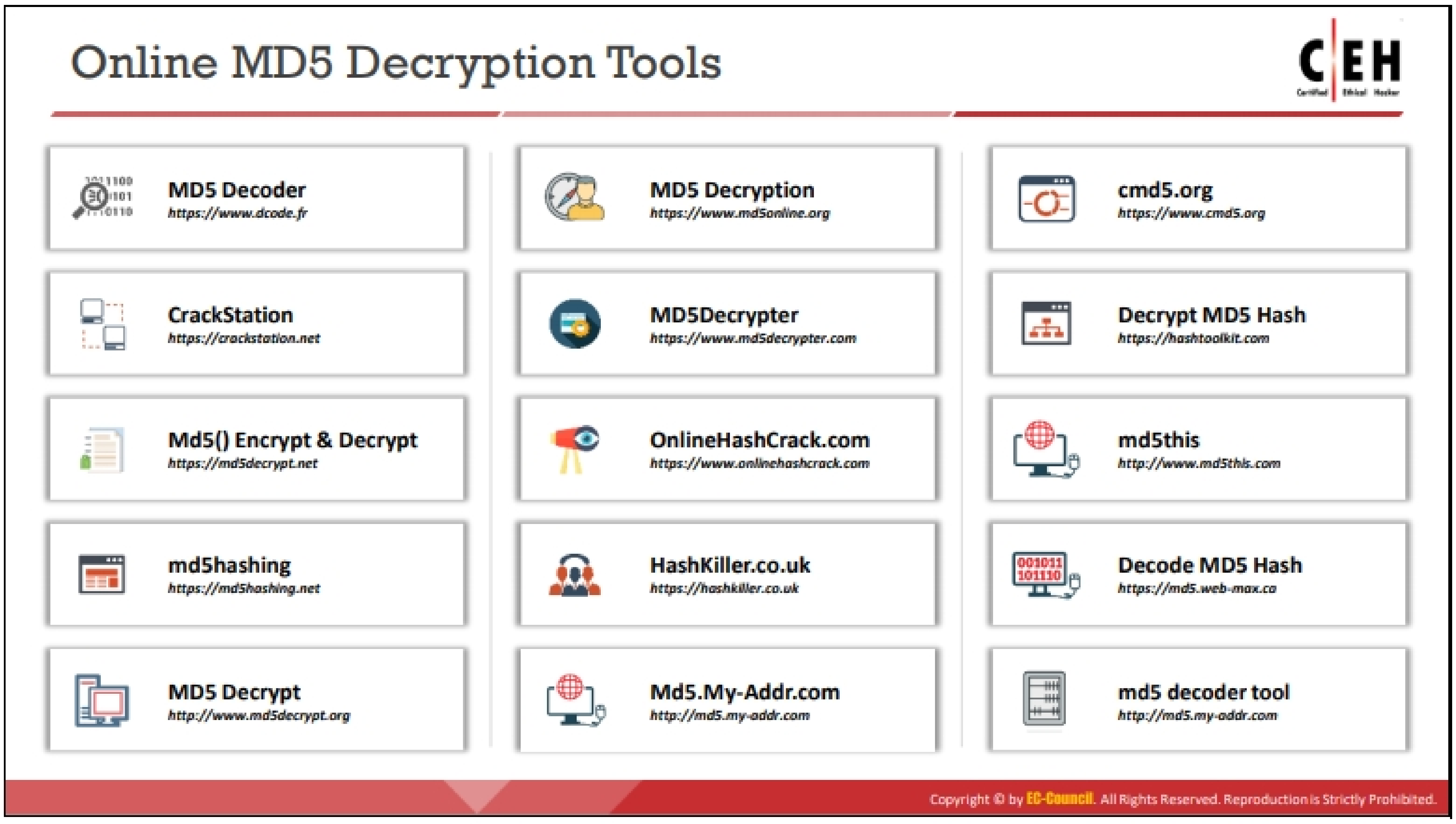
This text file was created in order to help you to n

1) The starting page of the online help offers the b
starting page you can reach all essential function
The starting page of the online help can be acces
top right of the main window or by using the search keyword "Starting page" within the index of

Figure 20.40: Screenshot of CrypTool

Some additional cryptanalysis tools are as follows:

- Cryptosense (*https://cryptosense.com*)
- RsaCtfTool (*https://github.com*)
- Msieve (*https://sourceforge.net*)
- Cryptol (*https://cryptol.net*)
- CryptoBench (*http://www.addario.org*)



Online MD5 Decryption Tools

MD5 Decoder
https://www.dcode.fr

CrackStation
https://crackstation.net

Md5() Encrypt & Decrypt
https://md5decrypt.net

md5hashing
https://md5hashing.net

MD5 Decrypt
http://www.md5decrypt.org

MD5 Decryption
https://www.md5online.org

MD5Decrypter
https://www.md5decrypter.com

OnlineHashCrack.com
https://www.onlinehashcrack.com

HashKiller.co.uk
https://hashkiller.co.uk

Md5.My-Addr.com
http://md5.my-addr.com

cmd5.org
https://www.cmd5.org

Decrypt MD5 Hash
https://hashtoolkit.com

md5this
http://www.md5this.com

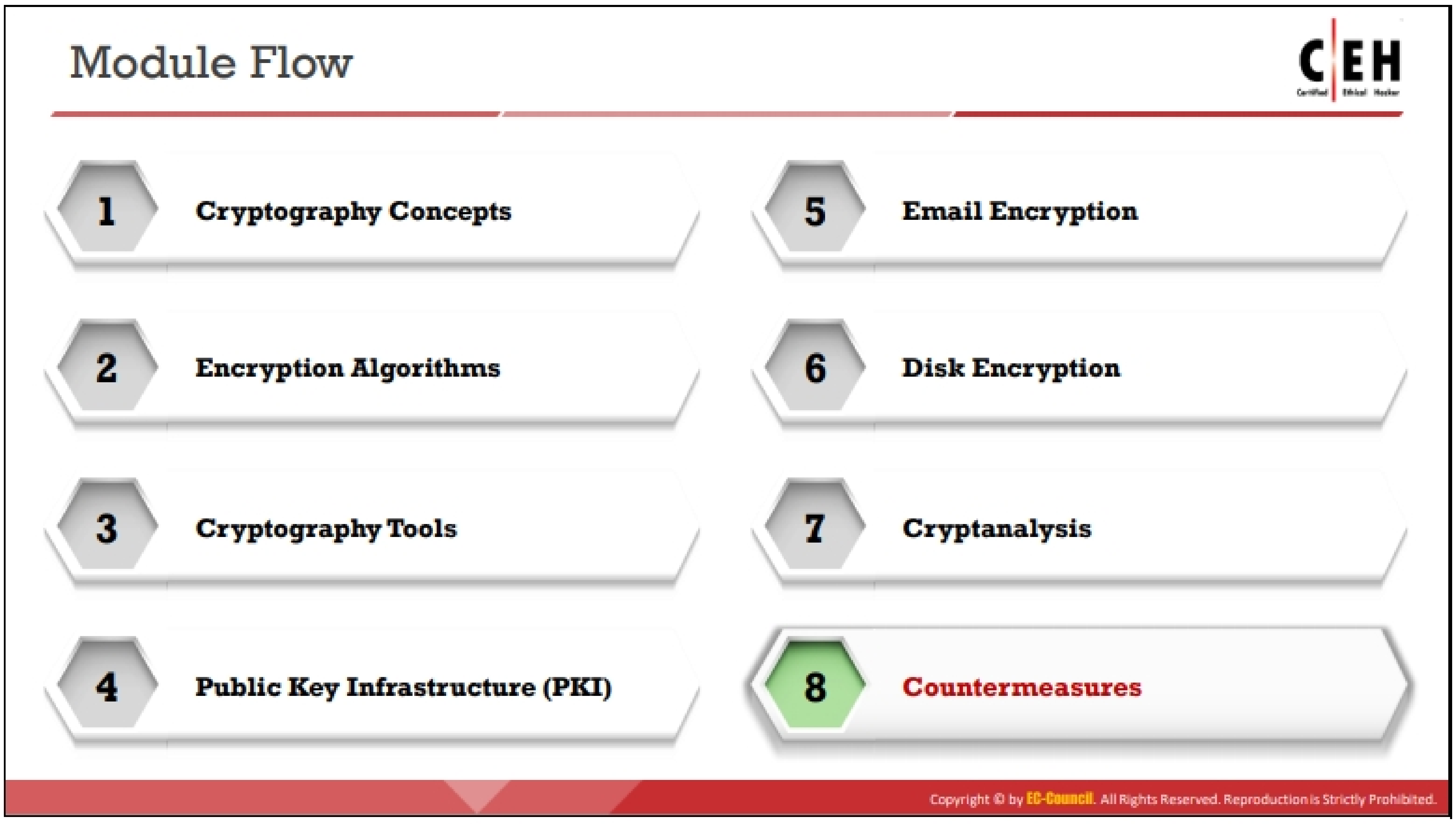
Decode MD5 Hash
https://md5.web-max.ca

md5 decoder tool
http://md5.my-addr.com

Online MD5 Decryption Tools

Some online MD5 decryption tools that can be used to decrypt the MD5 hash value to discover the original message are as follows:

- MD5 Decoder (*https://www.dcode.fr*)
- CrackStation (*https://crackstation.net*)
- Md5() Encrypt & Decrypt (*https://md5decrypt.net*)
- md5hashing (*https://md5hashing.net*)
- MD5 Decrypt (*http://www.md5decrypt.org*)
- MD5 Decryption (*https://www.md5online.org*)
- MD5Decrypter (*https://www.md5decrypter.com*)
- OnlineHashCrack.com (*https://www.onlinehashcrack.com*)
- HashKiller.co.uk (*https://hashkiller.co.uk*)
- Md5.My-Addr.com (*http://md5.my-addr.com*)
- cmd5.org (*https://www.cmd5.org*)
- md5this (*http://www.md5this.com*)
- Decode MD5 Hash (*https://md5.web-max.ca*)
- md5 decoder tool (*http://md5.my-addr.com*)

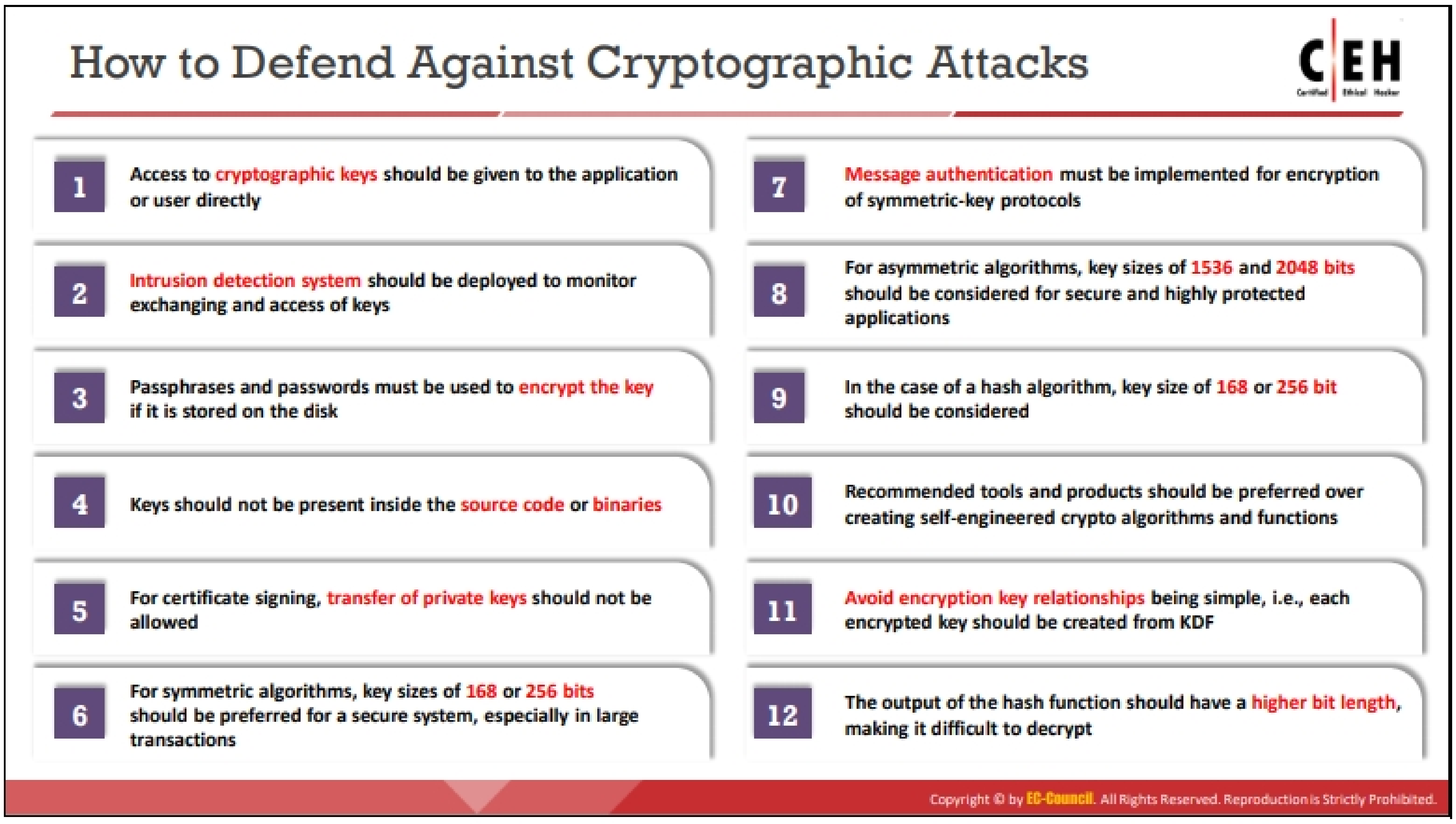


Module Flow

1. **Cryptography Concepts**
2. **Encryption Algorithms**
3. **Cryptography Tools**
4. **Public Key Infrastructure (PKI)**
5. **Email Encryption**
6. **Disk Encryption**
7. **Cryptanalysis**
8. **Countermeasures**

Countermeasures

Attackers use various cryptanalysis methods and techniques to break cryptosystems and steal confidential information that is transmitted in the network. This section discusses some countermeasures that can be adopted to prevent such attacks.



How to Defend Against Cryptographic Attacks

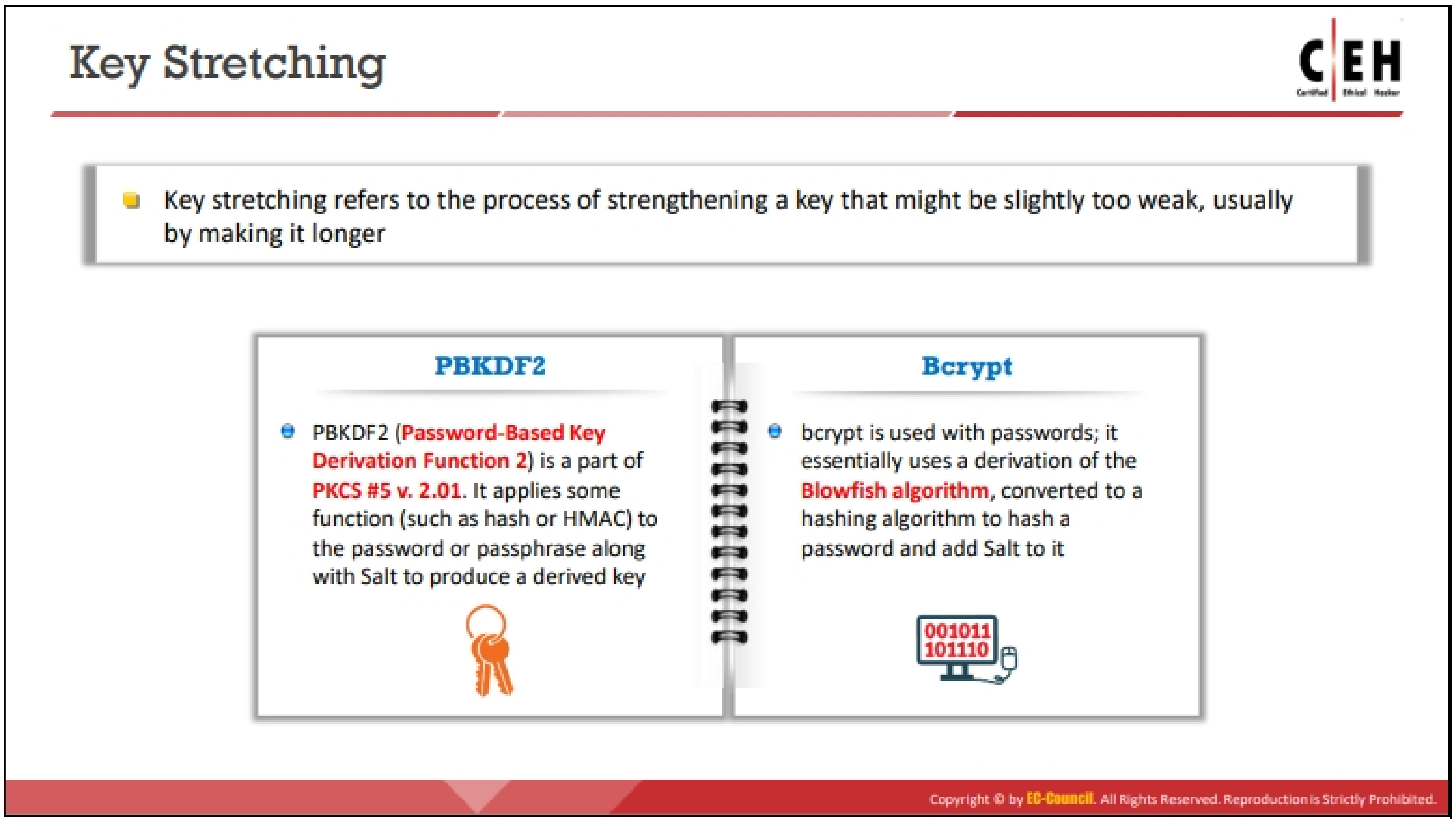
1. Access to cryptographic keys should be given to the application or user directly
2. Intrusion detection system should be deployed to monitor exchanging and access of keys
3. Passphrases and passwords must be used to encrypt the key if it is stored on the disk
4. Keys should not be present inside the source code or binaries
5. For certificate signing, transfer of private keys should not be allowed
6. For symmetric algorithms, key sizes of 168 or 256 bits should be preferred for a secure system, especially in large transactions
7. Message authentication must be implemented for encryption of symmetric-key protocols
8. For asymmetric algorithms, key sizes of 1536 and 2048 bits should be considered for secure and highly protected applications
9. In the case of a hash algorithm, key size of 168 or 256 bit should be considered
10. Recommended tools and products should be preferred over creating self-engineered crypto algorithms and functions
11. Avoid encryption key relationships being simple, i.e., each encrypted key should be created from KDF
12. The output of the hash function should have a higher bit length, making it difficult to decrypt

How to Defend Against Cryptographic Attacks

The following countermeasures can be adopted to prevent cryptographic attacks:

- Access of cryptographic keys should be given directly to the application or user.
- IDS should be deployed to monitor exchanging and access of keys.
- Passphrases and passwords must be used to encrypt the key, if stored in the disk.
- Keys should not be present inside the source code or binaries.
- For certificate signing, the transfer of private keys should not be allowed.
- For symmetric algorithms, a key size of 168 bits or 256 bits should be preferred for a secure system, especially in the case of large transactions.
- Message authentication must be implemented for the encryption of symmetric-key protocols.
- For asymmetric algorithms, a key size of 1536 bits or 2048 bits should be considered for secure and highly protected applications.
- In the case of hash algorithms, a key size of 168 or 256 bits should be considered.
- Only recommended tools or products should be used rather than self-engineered crypto algorithms or functions.
- Impose a limit on the number of operations per key.
- The output of the hash function should have a larger bit length that makes it difficult to decrypt.

- Design applications and protocols that can avoid simple encryption key relationships, i.e., each encrypted key should be created from a key derivation function (KDF).
- Upgrade to the latest security standards.
- Use strong key schedules to mitigate the risks of related key attacks.



Key Stretching

- Key stretching refers to the process of strengthening a key that might be slightly too weak, usually by making it longer

PBKDF2

- PBKDF2 (**Password-Based Key Derivation Function 2**) is a part of **PKCS #5 v. 2.01**. It applies some function (such as hash or HMAC) to the password or passphrase along with Salt to produce a derived key

Bcrypt

- bcrypt is used with passwords; it essentially uses a derivation of the **Blowfish algorithm**, converted to a hashing algorithm to hash a password and add Salt to it

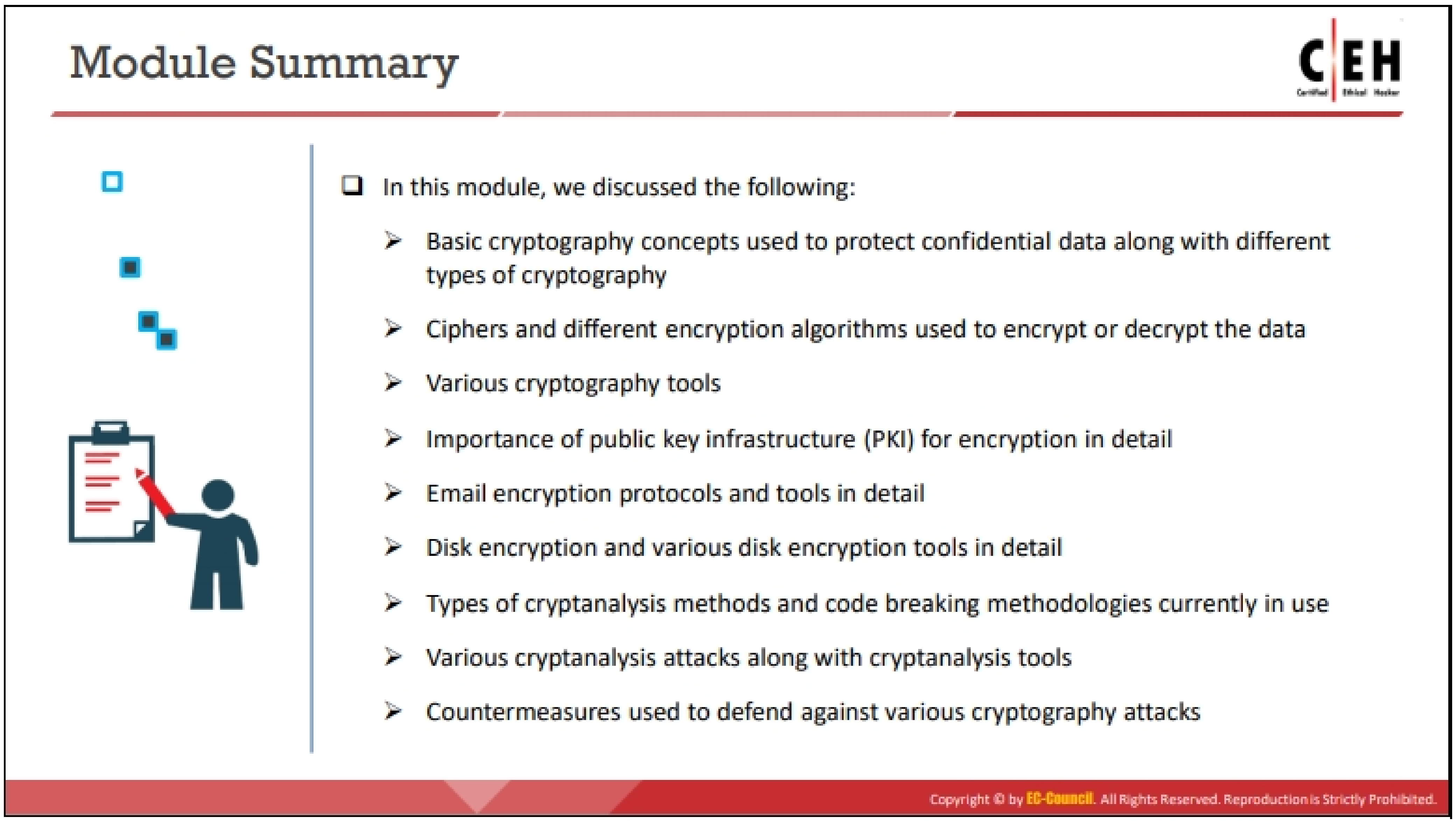
Key Stretching

Key stretching refers to processes used to make a weak key stronger, usually by making it longer. This technique helps in defending against brute-force attacks. In general, passwords or passphrases generated by end users are weak and predictable. Hence, key stretching helps security professionals/users to prevent such attacks by strengthening their passwords.

In the key stretching technique, the initial key is given as input to an algorithm that generates an enhanced key. The key must be sufficiently resistant to brute-force attacks. It is very difficult for the attackers to predict the enhanced key as they need to try every possible combination of the key or likely combinations of the enhanced key.

There are many functions and libraries that perform key stretching as part of their working:

- Password-Based Key Derivation Function 2 (PBKDF2) is part of PKCS #5 v. 2.01. It applies some function (such as a hash or HMAC) to the password or passphrase along with Salt to produce a derived key.
- bcrypt is used with passwords, and it essentially uses a variant of the Blowfish algorithm, converted to a hashing algorithm, to hash a password and add Salt to it.



Module Summary

☐ In this module, we discussed the following:

- Basic cryptography concepts used to protect confidential data along with different types of cryptography
- Ciphers and different encryption algorithms used to encrypt or decrypt the data
- Various cryptography tools
- Importance of public key infrastructure (PKI) for encryption in detail
- Email encryption protocols and tools in detail
- Disk encryption and various disk encryption tools in detail
- Types of cryptanalysis methods and code breaking methodologies currently in use
- Various cryptanalysis attacks along with cryptanalysis tools
- Countermeasures used to defend against various cryptography attacks

Module Summary

This module discussed basic cryptography concepts used to protect confidential data as well as different types of cryptography. It also described ciphers and different encryption algorithms used to encrypt or decrypt data in detail. Furthermore, it illustrated various cryptography tools. It then highlighted the importance of PKI in encryption and discussed the email encryption protocols and tools in detail. It also discussed disk encryption along with various disk encryption tools. Moreover, it explained the various types of cryptanalysis methods and code breaking methodologies in use. Subsequently, it presented the various types of cryptanalysis attacks and cryptanalysis tools. Finally, it ended with an explanation of the countermeasures against various cryptography attacks.

Glossary

A

- **Availability:** Assurance that the systems responsible for delivering, storing, and processing information are accessible when required by the authorized users.
- **Authenticity:** Refers to the characteristic of a communication, document, or any data that ensures the quality of being genuine.
- **Active Attacks:** Active attacks tamper with the data in transit or disrupt communication or services between the systems to bypass or break into secured systems.
- **Adversary Behavioral Identification:** Adversary behavioral identification involves the identification of the common methods or techniques followed by an adversary to launch attacks on or to penetrate an organization's network.
- **Active Footprinting:** Active footprinting involves gathering information about the target with direct interaction.
- **ARP Ping Scan:** Attackers send ARP request probes to target hosts, and an ARP response indicates that the host is active.
- **ACK Flag Probe Scan:** Attackers send TCP probe packets set with an ACK flag to a remote device, and then analyze the header information (TTL and WINDOW field) of received RST packets to determine if the port is open or closed.
- **Anonymizer:** An anonymizer is an intermediate server placed between you as the end user and the website to access the website on your behalf and make your web surfing activities untraceable.
- **Audio Steganography:** Audio steganography refers to hiding secret information in audio files such as .MP3, .RM, and .WAV.
- **Advanced Persistent Threats:** Advanced persistent threats (APTs) are defined as a type of network attack, where an attacker gains unauthorized access to a target network and remains undetected for a long period of time.
- **Antivirus Sensor System:** An antivirus sensor system is a collection of computer software that detects and analyzes malicious code threats such as viruses, worms, and Trojans.
- **Active Sniffing:** Active sniffing involves injecting Address Resolution Packets (ARP) into the network to flood the switch's Content Addressable Memory (CAM) table, which keeps track of host-port connections.
- **Address Resolution Protocol (ARP):** Address Resolution Protocol (ARP) is a stateless protocol used for resolving IP addresses to machine (MAC) addresses.
- **ARP Spoofing Attack:** ARP spoofing involves constructing many forged ARP request and reply packets to overload the switch.
- **Application Level Hijacking:** Application level hijacking refers to gaining control over the HTTP's user session by obtaining the session IDs.
- **Anomaly Detection:** It detects the intrusion based on the fixed behavioral characteristics of the users and components in a computer system.
- **Application-Level Firewall:** Application-level gateways (proxies) can filter packets at the application layer of the OSI model (or the application layer of TCP/IP).

- **Application Proxy:** An application-level proxy works as a proxy server and filters connections for specific services.
- **API DDoS Attack:** The DDoS attack involves saturating an API with a huge volume of traffic from multiple infected computers (botnet) to delay API services to legitimate users.
- **Automated Web App Security Testing:** It is a technique employed for automating the testing process. These testing methods and procedures are incorporated into each stage of development to report feedback constantly.
- **Application Whitelisting:** Application whitelisting contains a list of application components such as software libraries, plugins, extensions, and configuration files, which can be permitted to execute in the system.
- **Application Blacklisting:** Application blacklisting contains a list of malicious applications or software that are not permitted to be executed in the system or the network.
- **Access point (AP):** An AP is used to connect wireless devices to a wireless/wired network.
- **Association:** It refers to the process of connecting a wireless device to an AP.
- **Agent Smith Attack:** Agent Smith attacks are carried out by luring victims into downloading and installing malicious apps designed and published by attackers in the form of games, photo editors, or other attractive tools from third-party app stores such as 9Apps.
- **Android Rooting:** Rooting process involves exploiting security vulnerabilities in the device firmware and copying the SU binary to a location in the current process's PATH (e.g., /system/xbin/su) and granting it executable permissions with the chmod command.
- **Asymmetric Encryption:** Asymmetric encryption (public-key) uses different encryption keys, which are called public and private keys for encryption and decryption, respectively.
- **Advanced Encryption Standard (AES):** The Advanced Encryption Standard (AES) is a National Institute of Standards and Technology (NIST) specification for the encryption of electronic data.

B

- **Behavioral Indicators:** Behavioral indicators of compromise are used to identify specific behavior related to malicious activities.
- **Black Hats:** Black hats are individuals who use their extraordinary computing skills for illegal or malicious purposes.
- **BGP:** Border Gateway Protocol (BGP) is a routing protocol used to exchange routing and reachability information between different autonomous systems (AS) present on the Internet.
- **Brute-Force Attack:** In a brute-force attack, attackers try every combination of characters until the password is broken.
- **Buffer Overflow:** Buffer overflow or overrun is a common vulnerability in an applications or programs that accepts more data than the allocated buffer.
- **Baiting:** Baiting is a technique in which attackers offer end users something alluring in exchange for important information such as login details and other sensitive data.
- **Botnet:** A botnet is a huge network of compromised systems and can be used by an attacker to launch denial-of-service attacks.
- **Broken Access Control:** Broken access control is a method in which an attacker identifies a flaw related to access control and bypasses the authentication, which allows them to compromise the network.
- **Base64 Encoding:** The Base64 encoding scheme represents any binary data using only printable ASCII characters.

- **Bug Bounty Program:** The bug bounty program is a challenge hosted by organizations, websites, or software developers to tech-savvy individuals or ethical hackers to participate and break into their security to report the latest bugs and vulnerabilities.
- **Blind/Inferential SQL Injection:** In blind SQL injection, an attacker poses a true or false question to the database to determine whether the application is vulnerable to SQL injection.
- **Blacklist Validation:** Blacklist validation rejects all the malicious inputs that have been disapproved for protected access.
- **Bandwidth:** It describes the amount of information that may be broadcast over a connection.
- **Basic service set identifier (BSSID):** It is the media access control (MAC) address of an access point (AP) or base station that has set up a basic service set (BSS).
- **Bluetooth:** Bluetooth is a short-range wireless communication technology that replaces the cables connecting portable or fixed devices while maintaining high levels of security.
- **Bluesmacking:** A Bluesmacking attack occurs when an attacker sends an oversized ping packet to a victim's device, causing a buffer overflow.
- **BlueSniff:** BlueSniff is a proof-of-concept code for a Bluetooth wardriving utility.
- **BluePrinting:** BluePrinting is a footprinting technique performed by an attacker to determine the make and model of a target Bluetooth-enabled device.
- **Btlejacking:** A Btlejacking attack is detrimental to Bluetooth low energy (BLE) devices. The attacker can sniff, jam, and take control of the data transmission between BLE devices by performing an MITM attack.
- **Bluejacking:** Bluejacking is the activity of sending anonymous messages over Bluetooth to Bluetooth-enabled devices, such as laptop and mobile phones, via the OBEX protocol.
- **Bluesnarfing:** Bluesnarfing is the theft of information from a wireless device through a Bluetooth connection, often between phones, desktops, laptops, PDAs, and other devices.
- **Bluebugging:** Bluebugging involves gaining remote access to a target Bluetooth-enabled device and using its features without the victim’s knowledge or consent.
- **BYOD:** Bring your own device (BYOD) refers to a policy that allows an employee to bring their personal devices, such as laptops, smartphones, and tablets, to their workplace and use them to access the organization’s resources by following the access privileges.
- **BlueBorne Attack:** A BlueBorne attack is performed on Bluetooth connections to gain access and take full control of the target device.
- **Business Network:** It comprises of a network of systems that offer information infrastructure to the business.
- **Basic Process Control System (BPCS):** A BPCS is responsible for process control and monitoring of the industrial infrastructure.
- **Blowfish:** Blowfish is a type of symmetric block cipher algorithm designed to replace DES or IDEA algorithms.

C

- **Confidentiality:** Assurance that the information is accessible only to those authorized to have access.
- **Close-in Attacks:** Close-in attacks are performed when the attacker is in close physical proximity with the target system or network in order to gather, modify, or disrupt access to information.
- **Cyber Kill Chain Methodology:** The cyber kill chain methodology is a component of intelligence-driven defense for the identification and prevention of malicious intrusion activities.

- **Cyber Terrorists:** Cyber terrorists are individuals with a wide range of skills, motivated by religious or political beliefs, to create fear of large-scale disruption of computer networks.
- **Clearing Tracks:** Clearing tracks refers to the activities carried out by an attacker to hide malicious acts.
- **Cyber Threat Intelligence:** Cyber Threat Intelligence (CTI) is defined as the collection and analysis of information about threats and adversaries and the drawing of patterns that provide the ability to make knowledgeable decisions for preparedness, prevention, and response actions against various cyber-attacks.
- **Competitive Intelligence Gathering:** Competitive intelligence gathering is the process of identifying, gathering, analyzing, verifying, and using information about your competitors from resources, such as the Internet.
- **Common Vulnerability Scoring System (CVSS):** CVSS is a published standard that provides an open framework for communicating the characteristics and impacts of IT vulnerabilities.
- **Common Vulnerabilities and Exposures (CVE):** CVE® is a publicly available and free-to-use list or dictionary of standardized identifiers for common software vulnerabilities and exposures.
- **Common Weakness Enumeration (CWE):** Common Weakness Enumeration (CWE) is a category system for software vulnerabilities and weaknesses.
- **Combinator Attack:** Attackers combine the entries of the first dictionary with those of the second dictionary to generate a new wordlist to crack the password of the target system.
- **Crypter:** Software that protects malware from undergoing reverse engineering or analysis, thus making the task of the security mechanism harder in its detection.
- **Computer Worms:** Computer worms are malicious programs that independently replicate, execute, and spread across the network connections, thus consuming available computing resources without human interaction.
- **Chain Letters:** Emails that offer free gifts such as money and software on condition that the user forwards the mail to a specified number of people.
- **Compromised Insider:** An insider with access to critical assets of an organization who is compromised by an outside threat actor.
- **CRIME Attack:** Compression Ratio Info-Leak Made Easy (CRIME) is a client-side attack that exploits the vulnerabilities present in the data compression feature of protocols, such as SSL/TLS, SPDY, and HTTPS.
- **Circuit-Level Gateway Firewall:** Circuit-level gateways monitor requests to create sessions and determine if those sessions will be allowed.
- **Cross-Site Scripting (XSS) Attacks:** Cross-site scripting ('XSS' or 'CSS') attacks exploit vulnerabilities in dynamically generated web pages, enabling malicious attackers to inject client-side scripts into web pages viewed by other users.
- **Cross-Site Request Forgery (CSRF) Attack:** Cross-Site Request Forgery (CSRF) attacks exploit web page vulnerabilities that allow an attacker to force an unsuspecting user's browser to send malicious requests they did not intend.
- **Clickjacking Attack:** Attackers perform clickjacking attacks by tricking the victim into clicking on any malicious web page element that is placed transparently on the top of any trusted web page.
- **Cookie Poisoning:** It is a type of parameter tampering attack in which the attacker modifies the cookie contents to draw unauthorized information about a user and thus perform identity theft.
- **Cookie Sniffing:** It is a technique in which an attacker sniffs a cookie containing the session ID of the victim who has logged in to a target website and uses the cookie to bypass the authentication process and log in to the victim’s account.

- **Cookie Replay:** It is a technique used to impersonate a legitimate user by replaying the session/cookie that contains the session ID of that user (as long as he/she remains logged in).
- **Critical Infrastructure:** A collection of physical or logical systems and assets that the failure or destruction of which will severely impact the security, safety, economy, or public health.
- **Cloud Computing:** Cloud computing is an on-demand delivery of IT capabilities where IT infrastructure and applications are provided to subscribers as a metered service over a network.
- **Container-as-a-Service (CaaS):** It provides services such as virtualization of container engines, management of containers, applications, and clusters through a web portal, or an API.
- **Community Cloud:** It is a multi-tenant infrastructure shared among organizations from a specific community with common computing concerns, such as security, regulatory compliance, performance requirements, and jurisdiction.
- **Cloud Consumer:** A person or organization that uses cloud computing services.
- **Cloud Provider:** A person or organization providing services to interested parties.
- **Cloud Carrier:** An intermediary for providing connectivity and transport services between cloud consumers and providers.
- **Cloud Auditor:** A party for making independent assessments of cloud service controls and taking an opinion thereon.
- **Cloud Broker:** An entity that manages cloud services in terms of use, performance, and delivery, and maintains the relationship between cloud providers and consumers.
- **Container:** A container is a package of an application/software including all its dependencies such as library files, configuration files, binaries, and other resources that run independently of other processes in the cloud environment.
- **Container Orchestration:** Container orchestration is an automated process of managing the lifecycles of software containers and their dynamic environments.
- **Cloud Cryptojacking:** Cryptojacking is the unauthorized use of the victim's computer to stealthily mine digital currency.
- **Cloudborne Attack:** Cloudborne is a vulnerability residing in a bare-metal cloud server that enables the attackers to implant a malicious backdoor in its firmware.
- **Cloud Security Alliance (CSA):** CSA is a nonprofit global organization that provides rising awareness and promotes best practices and security policies to help and secure the cloud environment.
- **Cryptography:** Cryptography is the conversion of data into a scrambled code that is encrypted and sent across a private or public network.
- **CAST-128:** CAST-128, also called CAST5, is a symmetric-key block cipher having a classical 12- or 16-round Feistel network with a block size of 64 bits.
- **Camellia:** Camellia is a symmetric-key block cipher having either 18 rounds (for 128-bit keys) or 24 rounds (for 256-bit keys).
- **Cryptanalysis:** Cryptanalysis is the study of ciphers, ciphertext, or cryptosystems with the ability to identify vulnerabilities in them and thus extract plaintext from ciphertext even if the cryptographic key or algorithm used to encrypt the plaintext is unknown.

D

- **Distribution Attacks:** Distribution attacks occur when attackers tamper with hardware or software prior to installation.

- **Defense-in-Depth:** Defense-in-depth is a security strategy in which several protection layers are placed throughout an information system.
- **Deep web:** It consists of web pages and contents that are hidden and unindexed and cannot be located using traditional web browsers and search engines.
- **Dark Web or Darknet:** It is the subset of the deep web that enables anyone to navigate anonymously without being traced.
- **Dumpster Diving:** This uncouth technique, also known as trashing, involves the attacker rummaging for information in garbage bins.
- **DNS Cache Snooping:** DNS cache snooping is a DNS enumeration technique whereby an attacker queries the DNS server for a specific cached DNS record.
- **DNSSEC Zone Walking:** DNSSEC zone walking is a DNS enumeration technique where an attacker attempts to obtain internal records of the DNS server if the DNS zone is not properly configured.
- **Dictionary Attack:** In this type of attack, a dictionary file is loaded into a cracking application that runs against user accounts.
- **Distributed Network Attack:** A Distributed Network Attack (DNA) technique is used for recovering passwords from hashes or password-protected files using the unused processing power of machines across the network.
- **Document Steganography:** Document steganography is the technique of hiding secret messages transferred in the form of documents.
- **Downloader:** A type of Trojan that downloads other malware from the Internet on to the PC. Usually, attackers install downloader software when they first gain access to a system.
- **Dropper:** A type of Trojan that covertly installs other malware files on to the system.
- **Dynamic Malware Analysis:** It involves executing the malware code to know how it interacts with the host system and its impact on the system after infection.
- **DHCP Starvation Attack:** This is a denial-of-service (DoS) attack on the DHCP servers where the attacker broadcasts forged DHCP requests and tries to lease all the DHCP addresses available in the DHCP scope.
- **DNS Poisoning:** DNS poisoning is a technique that tricks a DNS server into believing that it has received authentic information when it has not received any
- **DNS Cache Poisoning:** DNS cache poisoning refers to altering or adding forged DNS records into the DNS resolver cache so that a DNS query is redirected to a malicious site.
- **Diversion Theft:** The attacker tricks a person responsible for making a genuine delivery into delivering the consignment to a location other than the intended location.
- **DoS Attack:** Denial-of-Service (DoS) is an attack on a computer or network that reduces, restricts, or prevents accessibility of system resources to its legitimate users.
- **DDoS Attack:** Distributed denial-of-service (DDoS) is a coordinated attack that involves a multitude of compromised systems (Botnet) attacking a single target, thereby denying service to users of the targeted system.
- **Distributed Reflection Denial-of-Service (DRDoS) Attack:** A distributed reflected denial-of-service attack (DRDoS), also known as a spoofed attack, involves the use of multiple intermediary and secondary machines that contribute to the actual DDoS attack against the target machine or application.
- **Demilitarized Zone (DMZ):** The demilitarized zone (DMZ) is an area that hosts computer(s) or a small sub-network placed as a neutral zone between a particular company's internal network and an untrusted external network to prevent outsider access to a company's private data.

- **Database Honeypots:** Database honeypots employ fake databases that are vulnerable to perform database-related attacks such as SQL injection and database enumeration.
- **DNS Server Hijacking:** Attacker compromises the DNS server and changes the DNS settings so that all the requests coming towards the target web server are redirected to his/her own malicious server.
- **Directory Traversal:** Directory traversal allows attackers to access restricted directories, including application source code, configuration, and critical system files to execute commands outside the web server's root application directory.
- **DNS Rebinding Attack:** Attackers use the DNS rebinding technique to bypass the same-origin policy’s security constraints, allowing the malicious web page to communicate with or make arbitrary requests to local domains.
- **Dynamic Application Security Testing (DAST):** It is also known as a black-box testing approach and is performed directly on running code to identify issues related to interfaces, requests/responses, sessions, scripts, authentication processes, code injections, etc.
- **Direct-Sequence Spread Spectrum (DSSS):** DSSS is a spread spectrum technique that multiplies the original data signal with a pseudo-random noise-spreading code.
- **Directional Antenna:** A directional antenna can broadcast and receive radio waves from a single direction.
- **Dipole Antenna:** A dipole antenna is a straight electrical conductor measuring half a wavelength from end to end, and it is connected at the center of the radio frequency (RF) feed line.
- **Distributed Control System (DCS):** DCS is a highly engineered and large-scale control system that is often used to perform industry specific tasks.
- **Docker:** Docker is an open source technology used for developing, packaging, and running applications and all its dependencies in the form of containers, to ensure that the application works in a seamless environment.
- **Data Encryption Standard (DES):** DES is designed to encipher and decipher blocks of data consisting of 64 bits under control of a 56-bit key.
- **DSA:** The Digital Signature Algorithm (DSA) is a Federal Information Processing Standard for digital signatures.
- **Diffie–Hellman:** It is a cryptographic protocol that allows two parties to establish a shared key over an insecure channel.
- **Digital Signature:** Digital signature uses asymmetric cryptography to simulate the security properties of a signature in digital rather than written form.
- **DUHK Attack:** DUHK (Don't Use Hard-Coded Keys) is a cryptographic vulnerability that allows an attacker to obtain encryption keys used to secure VPNs and web sessions.
- **DROWN Attack:** A DROWN attack is a cross-protocol weakness that can communicate and initiate an attack on servers that support recent SSLv3/TLS protocol suites.

E

- **Email Indicators:** Email indicators are used to send malicious data to the target organization or individual.
- **Ethical Hacking:** Ethical hacking involves the use of hacking tools, tricks, and techniques to identify vulnerabilities and ensure system security.
- **Eavesdropping:** Eavesdropping is the act of secretly listening to the conversations of people over a phone or video conference without their consent.
- **Enumeration:** Enumeration is the process of extracting usernames, machine names, network resources, shares, and services from a system or network.

- **Exploit:** A malicious code that breaches the system security via software vulnerabilities to access information or install malware.
- **Exploit Kit:** An exploit kit or crimeware toolkit is a platform to deliver exploits and payloads such as Trojans, spywares, backdoors, bots, and buffer overflow scripts to the target system.
- **Elicitation:** Attackers extract information from the victim by engaging him/her in normal and disarming conversations.
- **Egress Filtering:** Egress filtering scans the headers of IP packets leaving a network.
- **Email Honeypots:** Email honeypots are also called email traps. They are nothing but fake email addresses that are specifically used to attract fake and malicious emails from adversaries.
- **Error Based SQL Injection:** Error based SQL Injection forces the database to perform some operation in which the result will be an error.
- **Electronic Security Perimeter:** It is referred to as the boundary between secure and insecure zones.
- **Elliptic Curve Cryptography:** ECC is a modern public-key cryptography developed to avoid larger cryptographic key usage.

F

- **Federal Information Security Management Act (FISMA):** The FISMA provides a comprehensive framework for ensuring the effectiveness of information security controls over information resources that support Federal operations and assets.
- **Footprinting:** Footprinting is the first step of any attack on information systems in which an attacker collects information about a target network to identify various ways to intrude into the system.
- **Fingerprint Attack:** Attackers break down the passphrase into fingerprints comprising single and multi-character combinations to crack complex passwords.
- **Folder Steganography:** In folder steganography, files are hidden and encrypted within a folder and do not appear to normal Windows applications, including Windows Explorer.
- **Fileless Malware:** Fileless malware, also known as non-malware, infects legitimate software, applications, and other protocols existing in the system to perform various malicious activities.
- **File Fingerprinting:** File fingerprinting is the process of computing the hash value for a given binary code.
- **Forbidden Attack:** A forbidden attack is a type of man-in-the-middle attack used to hijack HTTPS sessions.
- **Firewall:** Firewalls are hardware and/or software designed to prevent unauthorized access to or from a private network.
- **Flooding:** The attacker sends loads of unnecessary traffic to produce noise, and if the IDS does not analyze the noise traffic well, then the true attack traffic may go undetected.
- **Firewalking:** Firewalking is a technique that uses TTL values to determine gateway ACL filters and it maps networks by analyzing the IP packet responses.
- **Frequency-Hopping Spread Spectrum (FHSS):** FHSS, also known as frequency-hopping code-division multiple access (FH-CDMA), is a method of transmitting radio signals by rapidly switching a carrier among many frequency channels.
- **Fault Injection Attacks:** Fault injection attacks, also known as Perturbation attacks, occur when a perpetrator injects any faulty or malicious program into the system to compromise the system security.
- **Function-as-a-Service (FaaS):** This cloud computing service provides a platform for developing, running, and managing application functionalities without the complexity of building and maintaining necessary infrastructure (serverless architecture).

G

- **Gray Hats:** Gray hats are the individuals who work both offensively and defensively at various times.
- **Gaining Access:** Gaining access refers to the point where the attacker obtains access to the operating system or applications on the target computer or network.
- **Google Hacking Database:** The Google Hacking Database (GHDB) is an authoritative source for querying the ever-widening reach of the Google search engine.
- **Global System for Mobile Communications (GSM):** It is a universal system used for mobile data transmission in wireless networks worldwide.
- **GOST Block Cipher:** The GOST (Government Standard) block cipher, also called Magma, is a symmetric-key block cipher having a 32-round Feistel network working on 64-bit blocks with a 256-bit key length.
- **GNU Privacy Guard:** GPG is a software replacement of PGP and free implementation of the OpenPGP standard.

H

- **Host-Based Indicators**: Host-based indicators are found by performing an analysis of the infected system within the organizational network.
- **Hacking:** Hacking refers to exploiting system vulnerabilities and compromising security controls to gain unauthorized or inappropriate access to a system's resources.
- **Hacker:** A hacker is a person who breaks into a system or network without authorization to destroy, steal sensitive data, or perform malicious attacks.
- **Hacktivist:** Individuals who promote a political agenda by hacking, especially by defacing or disabling websites.
- **Hash Injection/Pass-the-Hash (PtH) Attack:** A hash injection/PtH attack allows an attacker to inject a compromised hash into a local session and use the hash to validate network resources.
- **Host Integrity Monitoring:** Host integrity monitoring involves taking a snapshot of the system state using the same tools before and after analysis, to detect changes made to the entities residing on the system.
- **Hardware Protocol Analyzer:** A hardware protocol analyzer is a piece of equipment that captures signals without altering the traffic in a cable segment.
- **Honey Trap:** The honey trap is a technique where an attacker targets a person online by pretending to be an attractive person and then begins a fake online relationship to obtain confidential information about the target company.
- **Hoax Letters:** Emails that issue warnings to the user about new viruses, Trojans, or worms that may harm the user's system.
- **HTTP GET/POST Attack:** In an HTTP GET attack, attackers use a time-delayed HTTP header to maintain HTTP connections and exhaust web server resources.
- **HTTP Strict Transport Security (HSTS):** HTTP Strict Transport Security (HSTS) is a web security policy that protects HTTPS websites against MITM attacks.
- **HTTP Public Key Pinning (HPKP):** HTTP Public Key Pinning (HPKP) is a trust on first use (TOFU) technique used in an HTTP header that allows a web client to associate a specific public key certificate with a particular server to minimize the risk of MITM attacks based on fraudulent certificates.
- **Hardware Firewall:** A hardware firewall is either a dedicated stand-alone hardware device or it comes as part of a router.

- **Honeypot:** A honeypot is an information system resource that is expressly set up to attract and trap people who attempt to penetrate an organization's network.
- **High-Interaction Honeypots:** Unlike their low- and medium-interaction counterparts, high-interaction honeypots do not emulate anything; they run actual vulnerable services or software on production systems with real OS and applications.
- **Honeynets:** Honeynets are networks of honeypots. They are very effective in determining the entire capabilities of the adversaries.
- **HTTP Response-Splitting Attack:** An HTTP response-splitting attack is a web-based attack in which the attacker tricks the server by injecting new lines into response headers, along with arbitrary code.
- **Hotfixes:** Hotfixes are an update to fix a specific customer issue and not always distributed outside the customer organization.
- **HTML Encoding:** An HTML encoding scheme is used to represent unusual characters so that they can be safely combined within an HTML document.
- **Hex Encoding:** The HTML encoding scheme uses the hex value of every character to represent a collection of characters for transmitting binary data.
- **Hotspot:** Hotspots refer to areas with Wi-Fi availability, where users can enable Wi-Fi on their devices and connect to the Internet.
- **Hybrid Cloud:** It is a cloud environment comprised of two or more clouds (private, public, or community) that remain unique entities but are bound together to offer the benefits of multiple deployment models.
- **HMAC:** HMAC is a type of message authentication code (MAC) that combines a cryptographic key with a cryptographic hash function.
- **Homomorphic Encryption:** Homomorphic encryption allows users to secure and leave their data in an encrypted format even while it is being processed or manipulated.
- **Hardware-Based Encryption:** Hardware-based encryption uses computer hardware for assisting or replacing the software when the data encryption process is underway.
- **HSM:** Hardware security module (HSM) is an additional external security device that is used in a system for crypto-processing and can be used for managing, generating, and securely storing cryptographic keys.
- **Hard Drive Encryption:** Hard drive encryption is a technology where the data stored in the hardware can be encrypted using a wide range of encryption options.
- **Hash Collision Attack:** A hash collision attack is performed by finding two different input messages that result in the same hash output.

I

- **Integrity:** The trustworthiness of data or resources in terms of preventing improper or unauthorized changes.
- **Information Warfare:** The term information warfare or InfoWar refers to the use of information and communication technologies (ICT) to gain competitive advantages over an opponent.
- **Indicators of Compromise (IoCs):** Indicators of Compromise (IoCs) are the clues, artifacts, and pieces of forensic data found on the network or operating system of an organization that indicate a potential intrusion or malicious activity in the organization's infrastructure.
- **Information Assurance (IA):** IA refers to the assurance that the integrity, availability, confidentiality, and authenticity of information and information systems is protected during the usage, processing, storage, and transmission of information.

- **Incident Management:** Incident management is a set of defined processes to identify, analyze, prioritize, and resolve security incidents to restore normal service operations as quickly as possible and prevent future recurrence of the incident.
- **Incident Handling and Response:** Incident handling and response (IH&R) is the process of taking organized and careful steps when reacting to a security incident or cyberattack.
- **ISO/IEC 27001:2013:** ISO/IEC 27001:2013 specifies the requirements for establishing, implementing, maintaining, and continually improving an information security management system within the context of the organization.
- **Impersonation:** Pretending to be a legitimate or authorized person and using the phone or other communication medium to mislead targets and trick them into revealing information.
- **ICMP ECHO Ping Scan:** ICMP ECHO ping scans involve sending ICMP ECHO requests to a host. If the host is live, it will return an ICMP ECHO reply.
- **ICMP ECHO Ping Sweep:** Ping sweep is used to determine the live hosts from a range of IP addresses by sending ICMP ECHO requests to multiple hosts. If a host is alive, it will return an ICMP ECHO reply.
- **Inverse TCP Flag Scan:** Attackers send TCP probe packets with a TCP flag (FIN, URG, PSH) set or with no flags, where no response implies that the port is open, whereas an RST response means that the port is closed.
- **IP Address Decoy:** IP address decoy technique refers to generating or manually specifying the IP addresses of decoys in order to evade an IDS or firewall.
- **IP Address Spoofing:** IP spoofing refers to changing the source IP addresses so that the attack appears to be coming from someone else.
- **Image Steganography:** In image steganography, the information is hidden in image files of different formats such as .PNG, .JPG, and .BMP.
- **Injector:** A program that injects its code into other vulnerable running processes and changes how they execute to hide or prevent its removal.
- **IRDP Spoofing:** The attacker sends a spoofed IRDP router advertisement message to the host on the subnet, causing it to change its default router to whatever the attacker chooses.
- **Insider Attack:** An insider attack involves using privileged access to intentionally violate rules or cause threat to the organization's information or information systems in any form.
- **Identity Theft:** Identity theft is a crime in which an imposter steals your personally identifiable information such as name, credit card number, social security or driver's license numbers, etc. to commit fraud or other crimes.
- **ICMP Flood Attack:** ICMP flood attacks are a type of attack in which attackers send large volumes of ICMP echo request packets to a victim system directly or through reflection networks.
- **Ingress Filtering:** Ingress filtering prevents the source address spoofing of Internet traffic.
- **IPSec:** IPSec is a protocol suite developed by the IETF for securing IP communications by authenticating and encrypting each IP packet of a communication session.
- **Intrusion Detection System (IDS):** An intrusion detection system (IDS) is a software system or hardware device that inspects all inbound and outbound network traffic for suspicious patterns that may indicate a network or system security breach.
- **Intrusion Prevention System (IPS):** IPS are continuous monitoring systems that often sit behind firewalls as an additional layer of protection.

- **Insertion Attack:** Insertion is the process by which the attacker confuses the IDS by forcing it to read invalid packets.
- **Injection Flaws:** Injection flaws are web application vulnerabilities that allow untrusted data to be interpreted and executed as part of a command or query.
- **In-band SQL Injection:** An attacker uses the same communication channel to perform the attack and retrieve the results.
- **Input Validation:** Input validation helps developers to prevent user-supplied data influencing the logic of the code.
- **Industrial, Scientific, and Medical (ISM) Band:** This band is a set of frequencies used by the international industrial, scientific, and medical communities.
- **iOS Trustjacking:** iOS Trustjacking is a vulnerability that can be exploited by an attacker to read messages and emails and capture sensitive information from a remote location without the victim's knowledge.
- **IoT:** Internet of Things (IoT), also known as Internet of Everything (IoE), refers to the network of devices having IP addresses and the capability to sense, collect, and send data using embedded sensors, communication hardware and processors.
- **IoT Device Management:** IoT device management helps in supporting IoT solutions by using any software tools and processes and helps in onboarding latest devices securely and promptly.
- **Industrial Network:** A network of automated control systems is known as an industrial network.
- **Industrial Protocols:** Protocols used for serial communication and communication over standard Ethernet. Ex: S7, CDA, CIP, Modbus, etc.
- **IT/OT Convergence (IIOT):** IT/OT convergence is the integration of IT computing systems and OT operation monitoring systems to bridge the gap between IT/OT technologies for improving overall security, efficiency, and productivity.
- **ICS:** ICS is often referred to as a collection of different types of control systems and their associated equipment such as systems, devices, networks, and controls used to operate and automate several industrial processes.
- **Infrastructure-as-a-Service (IaaS):** This service provides virtual machines and other abstracted hardware and operating systems (OSs), which may be controlled through a service application programming interface (API).
- **Identity-as-a-Service (IDaaS):** This cloud computing service offers authentication services to the subscribed enterprises and is managed by a third-party vendor to provide identity and access management services.

J

- **Jailbreaking:** Jailbreaking is defined as the process of installing a modified set of kernel patches that allows users to run third-party applications not signed by the OS vendor.
- **Jamming Attack:** Jamming is a type of attack in which the communications between wireless IoT devices are jammed so that they can be compromised.

K

- **Kerberos:** Kerberos is a network authentication protocol that provides strong authentication for client/server applications through secret-key cryptography.
- **Keylogger:** Keystroke loggers are programs or hardware devices that monitor each keystroke as the user types on a keyboard, logs onto a file, or transmits them to a remote location.
- **KNOB attack:** A Key Negotiation of Bluetooth (KNOB) attack enables an attacker to breach Bluetooth security mechanisms and perform am MITM attack on paired devices without being traced.

- **Kubernetes:** Kubernetes, also known as K8s, is an open-source, portable, extensible, orchestration platform developed by Google for managing containerized applications and microservices.
- **Key Stretching:** Key stretching refers to the process of strengthening a key that might be slightly too weak, usually by making it longer.

L

- **LDAP:** Lightweight directory access protocol (LDAP) is an Internet protocol for accessing distributed directory services.
- **Lawful Interception:** Lawful interception refers to legally intercepting data communication between two end points for surveillance on the traditional telecommunications, Voice over Internet Protocol (VoIP), data, and multiservice networks.
- **Low-interaction Honeypots:** Low-interaction honeypots emulate only a limited number of services and applications of a target system or network.
- LDAP Injection Attack: An LDAP injection attack works in the same way as an SQL injection attack, but it exploits user parameters to generate an LDAP query.

M

- **Maintaining Access:** Maintaining access refers to the phase when the attacker tries to retain their ownership of the system.
- **Management Information Base (MIB):** MIB is a virtual database containing a formal description of all the network objects that can be managed using SNMP.
- **Markov-Chain Attack:** Attackers gather a password database and split each password entry into 2- and 3-character long syllables; using these character elements, a new alphabet is developed, which is then matched with the existing password database.
- **Malware:** Malware is malicious software that damages or disables computer systems and gives limited or full control of the systems to the malware creator for the purpose of theft or fraud.
- **Malicious Code:** A command that defines malware's basic functionalities such as stealing data and creating backdoors.
- **Malware Analysis:** Malware analysis is a process of reverse engineering a specific piece of malware to determine the origin, functionality, and potential impact of a given type of malware.
- **MAC Flooding:** MAC flooding involves the flooding of the CAM table with fake MAC address and IP pairs until it is full.
- **MAC Spoofing/Duplicating:** A MAC duplicating attack is launched by sniffing a network for MAC addresses of clients who are actively associated with a switch port and re-using one of those addresses.
- **Malicious Insider:** A disgruntled or terminated employee who steals data or destroys the company's networks intentionally by introducing malware into the corporate network.
- **Multi-Vector Attack:** In multi-vector DDoS attacks, the attackers use combinations of volumetric, protocol, and application-layer attacks to disable the target system or service.
- **Man-in-the-Middle Attack:** The man-in-the-middle attack is used to intrude into an existing connection between systems and intercept the messages being exchanged.
- **Man-in-the-Browser Attack:** The man-in-the-browser attack uses a Trojan horse to intercept the calls between the browser and its security mechanisms or libraries.
- **Medium-interaction Honeypots:** Medium-interaction honeypots simulate a real OS as well as applications and services of a target network.

- **Malware Honeypots:** Malware honeypots are used to trap malware campaigns or malware attempts over the network infrastructure.
- **MarioNet Attack:** MarioNet is a browser-based attack that runs malicious code inside the browser, and the infection persists even after closing or browsing away from the malicious webpage through which infection has spread.
- **Manual Web App Security Testing:** It involves testing a web application using manually designed data, customized code, and some browser extension tools to detect vulnerabilities and weaknesses associated with the applications.
- **Mobile Spam:** Mobile phone spam, also known as SMS spam, text spam, or m-spam, refers to unsolicited messages sent in bulk form to known/unknown phone numbers/email IDs to target mobile phones.
- **Mobile Device Management (MDM):** Mobile Device Management (MDM) provides platforms for over-the-air or wired distribution of applications and data and configuration settings for all types of mobile devices, including mobile phones, smartphones, and tablet computers.
- **Multi Cloud:** It is a dynamic heterogeneous environment that combines workloads across multiple cloud vendors that are managed via one proprietary interface to achieve long-term business goals.
- **Microservices:** Monolithic applications are broken down into cloud-hosted sub-applications called microservices that work together, each performing a unique task.
- **Man-in-the-Cloud (MITC) Attack:** MITC attacks are performed by abusing cloud file synchronization services such as Google Drive or Drop Box for Data compromise, command and control (C&C), data exfiltration, and remote access.
- **MD5:** The MD5 algorithm takes a message of arbitrary length as the input and then outputs a 128-bit fingerprint or message digest of the input.
- **MD6:** MD6 uses a Merkle-tree-like structure to allow for large-scale parallel computation of hashes for very long inputs.

N

- **Non-Repudiation:** A guarantee that the sender of a message cannot later deny having sent the message and that the recipient cannot deny having received the message.
- **Network Indicators:** Network indicators are useful for command and control, malware delivery, identifying the operating system, and other tasks.
- **Network Scanning:** Network scanning refers to a set of procedures used for identifying hosts, ports, and services in a network.
- **NTP:** Network Time Protocol (NTP) is designed to synchronize the clocks of networked computers.
- **National Vulnerability Database (NVD):** A U.S. government repository of standards-based vulnerability management data represented using the Security Content Automation Protocol (SCAP).
- **NTFS Data Stream:** NTFS Alternate Data Stream (ADS) is a Windows hidden stream, which contains metadata for the file, such as attributes, word count, author name and access, and modification time of the files.
- **Negligent Insider:** Insiders who are uneducated on potential security threats or who simply bypass general security procedures to meet workplace efficiency.
- **Network Level Hijacking:** Network level hijacking can be defined as the interception of packets during the transmission between a client and the server in a TCP or UDP session.
- **Network Address Translation (NAT):** Network address translation separates IP addresses into two sets and enables the LAN to use these addresses for internal and external traffic separately.

- **Network Perimeter:** It is the outermost boundary of a network zone i.e. closed group of assets.

O

- **OS Discovery/Banner Grabbing:** Banner grabbing or OS fingerprinting is the method used to determine the operating system running on a remote target system.
- **Obfuscator:** A program that conceals its code and intended purpose via various techniques, and thus, makes it hard for security mechanisms to detect or remove it.
- **Obfuscating:** Obfuscating is an IDS evasion technique used by attackers who encode the attack packet payload in such a way that the destination host can decode the packet but not the IDS.
- **OAuth:** OAuth is an authorization protocol that allows a user to grant limited access to their resources on a site to a different site without having to expose their credentials.
- **Output Encoding:** Output encoding is used to encode the input to ensure it is properly sanitized before being passed to the database.
- **Orthogonal Frequency-Division Multiplexing (OFDM):** An OFDM is a method of digital modulation of data in which a signal, at a chosen frequency, is split into multiple carrier frequencies that are orthogonal (occurring at right angles) to each other.
- **Omnidirectional Antenna:** Omnidirectional antennas radiate electromagnetic (EM) energy in all directions.
- **OT:** Operational Technology (OT) is the software and hardware designed to detect or cause changes in industrial operations through direct monitoring and/or controlling of industrial physical devices.
- **Operational Technology Cyber Security Alliance (OTCSA):** OTCSA educates operators and manufacturers with constant technical awareness and provides guidelines to apply essential changes, updates, integrations, etc.

P

- **Passive Attacks:** Passive attacks involve intercepting and monitoring network traffic and data flow on the target network and do not tamper with the data.
- **Procedures:** “Procedures” are organizational approaches that threat actors follow to launch an attack.
- **Payment Card Industry Data Security Standard (PCI DSS):** The Payment Card Industry Data Security Standard (PCI DSS) is a proprietary information security standard for organizations that handle cardholder information for major debit, credit, prepaid, e-purse, ATM, and POS cards.
- **Passive Footprinting:** Passive footprinting involves gathering information about the target without direct interaction.
- **Packet Fragmentation:** Packet fragmentation refers to the splitting of a probe packet into several smaller packets (fragments) while sending it to a network.
- **Proxy Server:** A proxy server is an application that can serve as an intermediary for connecting with other computers.
- **Password Cracking:** Password cracking techniques are used to recover passwords from computer systems.
- **Password Guessing:** Password guessing is a password-cracking technique that involves attempting to log on to the target system with different passwords manually.
- **Pass the Ticket Attack:** Pass the Ticket is a technique used for authenticating a user to a system that is using Kerberos without providing the user’s password.
- **PRINCE Attack:** An advanced version of a combinator attack where instead of taking input from two different dictionaries, attackers use a single input dictionary to build chains of combined words.

- **Password Salting:** Password salting is a technique where a random string of characters are added to the password before calculating their hashes.
- **Privilege Escalation:** A privilege escalation attack is the process of gaining more privileges than were initially acquired.
- **Packer:** A program that allows all files to bundle together into a single executable file via compression to bypass security software detection.
- **Payload:** A piece of software that allows control over a computer system after it has been exploited.
- **Packet Sniffing:** Packet sniffing is the process of monitoring and capturing all data packets passing through a given network using a software application or hardware device.
- **Passive Sniffing:** It involves monitoring packets sent by others without sending any additional data packets in the network traffic.
- **Piggybacking:** Piggybacking usually implies entry into a building or security area with the consent of the authorized person.
- **Pop-Up Windows**: Windows that suddenly pop up while surfing the Internet and ask for user information to login or sign-in.
- **Phishing:** Phishing is the practice of sending an illegitimate email claiming to be from a legitimate site in an attempt to acquire a user's personal or account information.
- **Pharming:** Pharming is a social engineering technique in which the attacker executes malicious programs on a victim's computer or server, and when the victim enters any URL or domain name, it automatically redirects the victim's traffic to an attacker-controlled website.
- **Professional Insider:** Harmful insiders who use their technical knowledge to identify weaknesses and vulnerabilities in the company's network and sell confidential information to competitors or black market bidders.
- **Ping of Death Attack:** In a Ping of Death (PoD) attack, an attacker tries to crash, destabilize, or freeze the targeted system or service by sending malformed or oversized packets using a simple ping command.
- **Pulse Wave DDoS Attack:** In a pulse wave DDoS attack, attackers send a highly repetitive, periodic train of packets as pulses to the target victim every 10 minutes, and each specific attack session can last for a few hours to days.
- **Peer-to-Peer Attack:** A peer-to-peer attack is a form of DDoS attack in which the attacker exploits a number of bugs in peer-to-peer servers to initiate a DDoS attack.
- **Permanent Denial-of-Service Attack:** Permanent DoS, also known as phlashing, refers to attacks that cause irreversible damage to system hardware.
- **Protocol Anomaly Detection:** In this type of detection, models are built to explore anomalies in the way in which vendors deploy the TCP/IP specification.
- **Packet Filtering Firewall:** In a packet filtering firewall, each packet is compared to a set of criteria before it is forwarded.
- **Pure Honeypots:** Pure honeypots emulate the real production network of a target organization.
- **Production Honeypots:** Production honeypots are deployed inside the production network of the organization along with other production servers.
- **Port Scanning:** Port scanning is used to identify open ports and the services running on these ports.
- **Patch:** A patch is a small piece of software designed to fix problems, security vulnerabilities, and bugs and improve the performance of a computer program or its supporting data.

- **Patch Management:** Patch management is a process used to fix known vulnerabilities by ensuring that the appropriate patches are installed on a system.
- **Parabolic Grid Antenna:** A parabolic grid antenna uses the same principle as a satellite dish, but it does not have a solid dish. It consists of a semi-dish in the form of a grid consisting of aluminum wires.
- **Purdue Model:** The Purdue model is derived from the Purdue Enterprise Reference Architecture (PERA) model, which is a widely used to describe internal connections and dependencies of important components in the ICS networks.
- **Programmable Logic Controller (PLC):** A programmable logic controller (PLC) is a small solid-state control computer where instructions can be customized to perform a specific task.
- **Platform-as-a-Service (PaaS):** This offers development tools, configuration management, and deployment platforms on-demand, which can be used by subscribers to develop custom applications.
- **Public Cloud:** In this model, the provider makes services such as applications, servers, and data storage available to the public over the Internet.
- **Private Cloud:** A private cloud, also known as the internal or corporate cloud, is a cloud infrastructure operated by a single organization and implemented within a corporate firewall.
- **Public Key Infrastructure (PKI):** PKI is a set of hardware, software, people, policies, and procedures required to create, manage, distribute, use, store, and revoke digital certificates.
- **Pretty Good Privacy (PGP):** It is often used for data compression, digital signing, encryption and decryption of messages, emails, files, and directories, and to enhance the privacy of email communications.
- **Padding Oracle Attack:** In a padding oracle attack (also known as a Vaudenay attack), attackers exploit the padding validation of an encrypted message to decipher the ciphertext.

Q

- **Quantum Cryptography:** This cryptography is processed based on quantum mechanics, such as quantum key distribution (QKD), using photons instead of mathematics as a part of encryption.

R

- **Reconnaissance:** Reconnaissance refers to the preparatory phase where an attacker seeks to gather information about a target prior to launching an attack.
- **Risk:** Risk refers to the degree of uncertainty or expectation that an adverse event may cause damage to the system.
- **Risk Management:** Risk management is the process of reducing and maintaining risk at an acceptable level by means of a well-defined and actively employed security program.
- **Risk Identification:** Identifies the sources, causes, consequences, and other details of the internal and external risks affecting the security of the organization.
- **Risk Assessment:** Assesses the organization's risk and provides an estimate of the likelihood and impact of the risk.
- **Risk Treatment:** Selects and implements appropriate controls for the identified risks.
- **Risk Tracking:** Ensures appropriate controls are implemented to handle known risks and calculates the chances of a new risk occurring.
- **Risk Review:** Evaluates the performance of the implemented risk management strategies.
- **RPC:** Remote Procedure Call (RPC) allows clients and servers to communicate in distributed client/server programs.

- **Replay Attack:** In a replay attack, packets and authentication tokens are captured using a sniffer. After the relevant information is extracted, the tokens are placed back on the network to gain access.
- **Rainbow Table:** A rainbow table is a precomputed table that contains word lists like dictionary files, brute force lists, and their hash values.
- **Rootkits:** Rootkits are programs that hide their presence as well as attacker's malicious activities, granting them full access to the server or host at that time, and in the future.
- **Ransomware:** Ransomware is a type of malware that restricts access to the computer system's files and folders and demands an online ransom payment to the malware creator(s) to remove the restrictions.
- **Rogue DHCP Server Attack:** The attacker sets up a rogue DHCP server on the network and responds to DHCP requests with bogus IP addresses resulting in compromised network access.
- **Reverse Social Engineering:** The attacker presents him/herself as an authority and the target seeks his or her advice before or after offering the information that the attacker needs.
- **RST Hijacking:** RST hijacking involves injecting an authentic-looking reset (RST) packet using a spoofed source address and predicting the acknowledgment number.
- **Research Honeypots:** Research honeypots are high-interaction honeypots primarily deployed by research institutes, governments, or military organizations to gain detailed knowledge about the actions of intruders.
- **RASP:** Runtime application self protection (RASP) provides security to web and non-web application running on a server.
- **Reflector Antennas:** Reflector antennas are used to concentrate EM energy that is radiated or received at a focal point.
- **Reverse Engineering:** Reverse engineering is the process of analyzing and extracting the source code of a software or application, and if needed, regenerating it with required modifications.
- **RC4:** RC4 is a variable key-size symmetric-key stream cipher with byte-oriented operations, and it is based on the use of a random permutation.
- **RC5:** RC5 is a fast symmetric-key block cipher designed by Ronald Rivest for RSA Data Security (now RSA Security).
- **RC6:** RC6 is a symmetric-key block cipher derived from RC5. It is a parameterized algorithm with a variable block size, key size, and number of rounds.
- **Rivest Shamir Adleman (RSA):** Ron Rivest, Adi Shamir, and Leonard Adleman formulated RSA, a public-key cryptosystem for Internet encryption and authentication.
- **RIPEMD-160:** RACE Integrity Primitives Evaluation Message Digest (RIPEMD) is a 160-bit hash algorithm developed by Hans Dobbertin, Antoon Bosselaers, and Bart Preneel.
- **Rainbow Table Attack:** A rainbow table attack is a type of cryptography attack where an attacker uses a rainbow table to reverse cryptographic hash functions.
- **Related-Key Attack:** An attacker launch a related key attack by exploiting the mathematical relationship between keys in a cipher to gain access over encryption and decryption functions.

S

- **Suicide Hackers:** Suicide hackers are individuals who aim to bring down critical infrastructure for a "cause" and are not worried about facing jail terms or any other kind of punishment.
- **Script Kiddies:** Script kiddies are unskilled hackers who compromise systems by running scripts, tools, and software developed by real hackers.

- **State-Sponsored Hackers:** State-sponsored hackers are individuals employed by the government to penetrate, gain top-secret information from, and damage the information systems of other governments.
- **Scanning:** Scanning refers to the pre-attack phase when the attacker scans the network for specific information based on information gathered during reconnaissance.
- **Supervised Learning:** Supervised learning uses algorithms that input a set of labeled training data to attempt to learn the differences between the given labels.
- **Sarbanes Oxley Act (SOX):** Enacted in 2002, the Sarbanes-Oxley Act aims to protect the public and investors by increasing the accuracy and reliability of corporate disclosures.
- **Shoulder Surfing:** In the shoulder surfing technique, an attacker stands behind the victim and secretly observes the victim's activities on the computer, such as keystrokes while entering usernames, passwords, and so on.
- **Stealth Scan (Half-open Scan):** Stealth scanning involves abruptly resetting the TCP connection between the client and server before the completion of three-way handshake signals, thus leaving the connection half-open.
- **SCTP INIT Scanning:** Attackers send an INIT chunk to the target host, and an INIT+ACK chunk response implies that the port is open, whereas an ABORT Chunk response means that the port is closed.
- **SCTP COOKIE ECHO Scanning:** Attackers send a COOKIE ECHO chunk to the target host, and no response implies that the port is open, whereas an ABORT Chunk response means that the port is closed.
- **Source Routing:** Source routing refers to sending a packet to the intended destination with a partially or completely specified route (without firewall-/IDS-configured routers) in order to evade an IDS or firewall.
- **Source Port Manipulation:** Source port manipulation refers to manipulating actual port numbers with common port numbers in order to evade an IDS or firewall.
- **SNMP Enumeration:** SNMP enumeration is the process of enumerating user accounts and devices on a target system using SNMP.
- **Spyware:** Spyware is a stealthy program that records the user's interaction with the computer and the Internet without the user's knowledge and sends the information to the remote attackers.
- **Steganography:** Steganography is a technique of hiding a secret message within an ordinary message and extracting it at the destination to maintain confidentiality of data.
- **Spam/Email Steganography:** Spam/email steganography refers to the technique of sending secret messages by hiding them in spam/email messages.
- **Steganalysis:** Steganalysis is the art of discovering and rendering covert messages using steganography.
- **Sheep Dip Computer:** Sheep dipping refers to the analysis of suspect files, incoming messages, etc. for malware.
- **Static Malware Analysis:** It involves going through the executable binary code without executing it to have a better understanding of the malware and its purpose.
- **System Baselining:** Baselining refers to the process of capturing the system state (taking a snapshot of the system) when the malware analysis begins, which can be compared with the system's state after executing the malware file.
- **SPAN Port:** A SPAN port is a port that is configured to receive a copy of every packet that passes through a switch.
- **STP Attack:** Attackers connect a rogue switch into the network to change the operations of the STP protocol and sniff all the network traffic.
- **Social Engineering:** Social engineering is the art of convincing people to reveal confidential information.

- **Spam Email:** Irrelevant, unwanted, and unsolicited emails that attempt to collect financial information, social security numbers, and network information.
- **Scareware:** Malware that tricks computer users into visiting malware infested websites, or downloading/buying potentially malicious software.
- **Spear Phishing:** Attackers send spear phishing to send a message with specialized, social engineering content directed at a specific person, or a small group of people.
- **Spimming:** A variant of spam that exploits Instant Messaging platforms to flood spam across the networks.
- **SMiShing:** SMiShing (SMS phishing) is the act of using SMS text messaging system of cellular phones or other mobile devices to lure users into instant action, such as downloading malware, visiting a malicious webpage, or calling a fraudulent phone number.
- **Smurf Attack:** In a Smurf attack, the attacker spoofs the source IP address with the victim's IP address and sends a large number of ICMP ECHO request packets to an IP broadcast network.
- **SYN Flood Attack:** In a SYN attack, the attacker sends a large number of SYN requests to the target server (victim) with fake source IP addresses.
- **Spoofed Session Flood Attack:** Attackers create fake or spoofed TCP sessions by carrying multiple SYN, ACK, and RST or FIN packets.
- **Session Hijacking:** Session hijacking refers to an attack in which an attacker seizes control of a valid TCP communication session between two computers.
- **Signature Recognition:** Signature recognition, also known as misuse detection, tries to identify events that indicate an abuse of a system or network resource.
- **Software Firewall:** A software firewall is a software program installed on a computer, just like normal software.
- **Stateful Multilayer Inspection Firewall:** Stateful multilayer inspection firewalls combine the aspects of the other three types of firewalls (Packet Filtering, Circuit-Level Gateways, and Application-Level Firewalls).
- **Spam Honeypots:** Spam honeypots specifically target spammers who abuse vulnerable resources such as open mail relays and open proxies.
- **Spider Honeypots:** Spider honeypots are also called spider traps. These honeypots are specifically designed to trap web crawlers and spiders.
- **Session Splicing:** Session splicing is a technique used to bypass the IDS where an attacker splits the attack traffic into many packets such that no single packet triggers the IDS.
- **Static Application Security Testing (SAST):** It is also referred to as a white-box testing approach, in which the complete system architecture (including its source code) or application/software to be tested is already known to the tester.
- **Source Code Review:** Source code reviews are used to detect bugs and irregularities in the developed web applications.
- **16-bit Unicode Encoding:** It replaces unusual Unicode characters with "%u" followed by the character's Unicode code point expressed in hexadecimal.
- **SQL Injection:** SQL injection is a technique used to take advantage of un-sanitized input vulnerabilities to pass SQL commands through a web application for execution by a backend database.
- **Service Set Identifier (SSID):** An SSID is a 32-alphanumeric-character unique identifier given to a wireless local area network (WLAN) that acts as a wireless identifier of the network.
- **Simjacker:** Simjacker is a vulnerability associated with a SIM card's S@T browser (SIMalliance Toolbox Browser), a pre-installed software incorporated in SIM cards to provide a set of instructions.

- **Sybil Attack:** The attacker uses multiple forged identities to create a strong illusion of traffic congestion, affecting communication between neighboring nodes and networks.
- **Side-Channel Attack:** The attacker extracts information about encryption keys by observing the emission of signals i.e. "side channels" from IoT devices.
- **Supervisory Control and Data Acquisition (SCADA):** SCADA is a centralized supervisory control system that is used for controlling and monitoring industrial facilities and infrastructure.
- **Safety Instrumented Systems (SIS):** An SIS is an automated control system designed to safeguard the manufacturing environment in case of any hazardous incident in the industry.
- **Software-as-a-Service (SaaS):** This cloud computing service offers application software to subscribers on-demand over the Internet.
- **Security-as-a-Service (SECaaS):** It provides services such as penetration testing, authentication, intrusion detection, anti-malware, security incident and event management.
- **Serverless Computing:** Serverless computing also known as serverless architecture or Function-as-a-Service (FaaS), is a cloud-based application architecture where application infrastructure and supporting services are provided by the cloud vendor as they are needed.
- **Symmetric Encryption:** Symmetric encryption (secret-key, shared-key, and private-key) uses the same key for encryption as it does for decryption.
- **Serpent:** Serpent uses a 128-bit symmetric block cipher with 128-, 192-, or 256-bit key sizes.
- **Secure Hashing Algorithm (SHA):** This algorithm generates a cryptographically secure one-way hash; it was published by the National Institute of Standards and Technology as a US Federal Information Processing Standard.
- **Secure Sockets Layer (SSL):** SSL is an application layer protocol developed by Netscape for managing the security of message transmission on the Internet.

T

- **Tactics, Techniques, and Procedures (TTPs):** The term Tactics, Techniques, and Procedures (TTPs) refers to the patterns of activities and methods associated with specific threat actors or groups of threat actors.
- **Tactics:** “Tactics” are the guidelines that describe the way an attacker performs the attack from beginning to the end.
- **Techniques:** “Techniques” are the technical methods used by an attacker to achieve intermediate results during the attack.
- **Threat Modeling:** Threat modeling is a risk assessment approach for analyzing the security of an application by capturing, organizing, and analyzing all the information that affects the security of an application.
- **The Digital Millennium Copyright Act (DMCA):** It defines the legal prohibitions against the circumvention of technological protection measures employed by copyright owners to protect their works, and against the removal or alteration of copyright management information.
- **Traceroute:** Traceroute programs work on the concept of ICMP protocol and use the TTL field in the header of ICMP packets to discover the routers on the path to a target host.
- **Toggle-Case Attack:** Attackers try all possible combinations of upper and lower cases of a word present in the input dictionary.
- **Trojan:** It is a program in which the malicious or harmful code is contained inside apparently harmless programming or data in such a way that the code can get control and cause damage, such as ruining the file allocation table on your hard disk.

- **Tailgating:** Tailgating implies accessing a building or secured area without the consent of the authorized person.
- **Throttling:** Throttling entails the setting up of routers for server access with a logic to throttle incoming traffic levels that are safe for the server.
- **TCP/IP Hijacking:** TCP/IP hijacking involves using spoofed packets to seize control of a connection between a victim and target machine.
- **Two-Factor Authentication:** A two-factor authentication provides an extra layer of protection as it provides another vector of authentication in addition to a user's password.
- **Triple Data Encryption Standard (3DES):** It performs DES three times with three different keys.
- **Twofish:** Twofish uses a block size of 128 bits and key sizes up to 256 bits. It is a Feistel cipher.
- **Threefish:** Threefish is a large tweakable symmetric-key block cipher in which the block and key sizes are equal, i.e., 256, 512, and 1024.
- **TEA:** Tiny Encryption Algorithm (TEA) is a Feistel cipher that uses 64 rounds.
- **TPM:** Trusted platform module (TPM) is a crypto-processor or chip that is present on the motherboard that can securely store the encryption keys, and it can perform many cryptographic operations.
- **Transport Layer Security (TLS):** TLS is a protocol to establish a secure connection between a client and a server and ensure the privacy and integrity of information during transmission.

U

- **Unsupervised Learning:** Unsupervised learning makes use of algorithms that input unlabeled training data to attempt to deduce all the categories without guidance.
- **UDP Ping Scan:** Attackers send UDP packets to target hosts, and a UDP response indicates that the host is active.
- **UDP Flood Attack:** An attacker sends spoofed UDP packets at a very high packet rate to a remote host on random ports of a target server using a large source IP range.
- **UDP Hijacking:** A network-level session hijacking where the attacker sends forged server reply to a victim's UDP request before the intended server replies to it.
- **URL Encoding:** URL encoding is the process of converting URL into valid ASCII format so that data can be safely transported over HTTP.
- **UTF-8:** It is a variable-length encoding standard that uses each byte expressed in hexadecimal and preceded by the % prefix.
- **Union SQL Injection:** In a UNION SQL injection, an attacker combines a forged query with a query requested by the user using a UNION clause.
- **USB Encryption:** USB encryption is an additional feature for USB storage devices that offers onboard encryption services.

V

- **Vulnerability Research:** Vulnerability research is the process of analyzing protocols, services, and configurations to discover the vulnerabilities and design flaws that will expose an operating system and its applications to exploit, attack, or misuse.
- **Vulnerability Assessment:** Vulnerability assessment is an in-depth examination of the ability of a system or application, including current security procedures and controls, to withstand the exploitation.
- **Vulnerability Exploitation:** Vulnerability exploitation involves the execution of multiple complex, interrelated steps to gain access to a remote system.

- **Video Steganography:** Video steganography refers to hiding secret information in a carrier video file.
- **Virus:** A virus is a self-replicating program that produces its own copy by attaching itself to another program, computer boot sector or document.
- **Vishing:** Vishing (voice or VoIP phishing) is an impersonation technique (electronic fraud) in which the attacker tricks individuals to reveal personal and financial information using voice technology such as the telephone system, VoIP, etc.
- **VPN:** A VPN is a private network constructed using public networks, such as the Internet.
- **Vulnerability Scanning:** Vulnerability scanning is performed to identify vulnerabilities and misconfigurations in a target web server or network.

W

- **White Hats:** White hats or penetration testers are individuals who use their hacking skills for defensive purposes.
- **Website Footprinting:** Website footprinting refers to the monitoring and analysis of the target organization’s website for information.
- **Whois:** Whois is a query and response protocol used for querying databases that store the registered users or assignees of an Internet resource, such as a domain name, an IP address block, or an autonomous system.
- **Wire Sniffing:** Packet sniffing is a form of wire sniffing or wiretapping in which hackers sniff credentials during transit by capturing Internet packets.
- **Windows Management Instrumentation (WMI):** WMI is a feature in Windows administration that provides a platform for accessing Windows system resources locally and remotely.
- **Windows Remote Management (WinRM):** WinRM is a Windows-based protocol designed to allow a user to run an executable file, modify system services, and the registry on a remote system.
- **Whitespace Steganography:** In white space steganography, the user hides the messages in ASCII text by adding white spaces to the ends of the lines.
- **Wiretapping:** Wiretapping is the process of the monitoring of telephone and Internet conversations by a third party.
- **Whaling:** A whaling attack is a type of phishing that targets high profile executives like CEO, CFO, politicians, and celebrities who have complete access to confidential and highly valuable information.
- **Web Server:** A web server is a computer system that stores, processes, and delivers web pages to clients via HTTP.
- **Website Defacement:** Website defacement refers to unauthorized changes made to the content of a single web page or an entire website, resulting in changes to the visual appearance of the web page or website.
- **Web Server Misconfiguration:** Server misconfiguration refers to configuration weaknesses in web infrastructure that can be exploited to launch various attacks on web servers such as directory traversal, server intrusion, and data theft.
- **Website Mirroring:** Website mirroring copies an entire website and its content onto a local drive.
- **Web Applications:** Web applications provide an interface between end users and web servers through a set of web pages that are generated at the server end or contain script code to be executed dynamically within the client web browser.
- **Web Service:** A web service is an application or software that is deployed over the Internet and uses standard messaging protocols such as SOAP, UDDI, WSDL, and REST to enable communication between applications developed for different platforms.

- **Web-based Timing Attack:** A web-based timing attack is a type of side-channel attack performed by attackers to retrieve sensitive information such as passwords from web applications by measuring the response time taken by the server.
- **Web Spidering/Crawling:** Web spiders/crawlers automatically discover the hidden content and functionality by parsing HTML forms and client-side JavaScript requests and responses.
- **WS-Address Spoofing:** In a WS-address spoofing attack, an attacker sends a SOAP message containing fake WS-address information to the server. The <ReplyTo> header consists of the address of the endpoint selected by the attacker rather than the address of the web service client.
- **Web API:** Web API is an application programming interface that provides online web services to client-side apps for retrieving and updating data from multiple online sources.
- **Webhooks:** Webhooks are user-defined HTTP callback or push APIs that are raised based on events triggered, such as receiving a comment on a post or pushing code to the registry.
- **Web Shell:** A web shell is a malicious piece of code or script that is developed using server-side languages such as PHP, ASP, PERL, RUBY, and Python and are then installed on a target server.
- **Web Application Fuzz Testing:** Web application fuzz testing (fuzzing) is a black-box testing method. It is a quality checking and assurance technique used to identify coding errors and security loopholes in web applications.
- **Whitelist Validation:** Whitelist validation is an effective technique in which only the list of entities that have been approved for secured access are accepted.
- **Wi-Fi:** Wireless network (Wi-Fi) refers to WLANs based on IEEE 802.11 standard, which allows the device to access the network from anywhere within an AP range.
- **Wired Equivalent Privacy (WEP):** WEP is a security protocol defined by the 802.11b standard; it was designed to provide a wireless LAN with a level of security and privacy comparable to that of a wired LAN.
- **Wi-Fi Protected Access (WPA):** WPA is a security protocol defined by 802.11i standards; it uses a Temporal Key Integrity Protocol (TKIP) that utilizes the RC4 stream cipher encryption with 128-bit keys and 64-bit MIC integrity check to provide stronger encryption and authentication.
- **WPA2:** WPA2 is an upgrade to WPA, and it includes mandatory support for counter mode with cipher block chaining message authentication code protocol (CCMP), an AES-based encryption mode with strong security.
- **WPA3:** WPA3 is an advanced implementation of WPA2 providing trailblazing protocols and uses the AES-GCMP 256 encryption algorithm.
- **Wireless Traffic Analysis:** Wireless traffic analysis enables attackers to identify vulnerabilities and susceptible victims in a target wireless network.
- **Wireless Intrusion Prevention Systems:** Wireless intrusion prevention systems (IPSs) protect networks against wireless threats and enable administrators to detect and prevent various network attacks.
- **Wrapping Attack:** A wrapping attack is performed during the translation of the SOAP message in the TLS layer where attackers duplicate the body of the message and sends it to the server as a legitimate user.
- **Web of Trust (WOT):** Web of trust (WoT) is a trust model of PGP, OpenPGP, and GnuPG systems.

X

- **Xmas Scan:** Xmas scan is a type of inverse TCP scanning technique with the FIN, URG, and PUSH flags set to send a TCP frame to a remote device.

- **XML External Entity Attack:** XML External Entity attack is a server-side request forgery (SSRF) attack that can occur when a misconfigured XML parser allows applications to parse XML input from an unreliable source.

Y

- **Yagi Antenna:** A Yagi antenna, also called Yagi–Uda antenna, is a unidirectional antenna commonly used in communications at a frequency band of 10 MHz to VHF and UHF.
- **YAK:** YAK is a public-key-based Authenticated Key Exchange (AKE) protocol.

Z

- **Zones and Conduits:** A network segregation technique used to isolate the networks and assets to impose and maintain strong access control mechanisms.
- **Zero Trust Network:** The Zero Trust model is a security implementation that assumes that every user trying to access the network is not a trusted entity by default and verifies every incoming connection before allowing access to the network.

References

Module 01: Introduction to Ethical Hacking

1. (2006), Ethical Hacking, from http://neworder.box.sk/news/921.
2. (2006), Hacker methodology, from http://www.hackersecuritymeasures.com/.
3. (2006), The Cybercrime Act 2001 Australia, Germany, Singapore Chapter 50A: Computer misuse Act, from http://www.cybercrimelaw.net/laws/countries/australia.html, http://www.cybercrimelaw.net/laws/countries/germany.html, http://www.mosstingrett.no/info/legal.html#29.
4. (2006), Computer Misuse Act 1990 Chapter 18 Unauthorized access to computer material, from http://www.cybercrimelaw.net/laws/countries/uk.html.
5. Police and Justice Act 2006, http://www.opsi.gov.uk/acts/acts2006/ukpga_20060048_en_7#pt5-pb2.
6. Dan Goodin, (2012), Zero-day attacks are meaner, more rampant than we ever thought, from https://arstechnica.com/information-technology/2012/10/zero-day-attacks-are-meaner-and-more-plentiful-than-thought/.
7. Frank T. Bass, (1998), SECURITY POLICY: TARGET, CONTENT, & LINKS, from https://csrc.nist.gov/csrc/media/publications/conference-paper/1998/10/08/proceedings-of-the-21st-nissc-1998/documents/paperg4.pdf.
8. (2009,)Anatomy of the Hack - Hands-on Security, from https://www.slideshare.net/NewBU/anatomy-of-the-hack-handson-security-information-assurance-club.
9. Nick Sutton and Jonathan Knight, (2014), Online In 60 Seconds – A Year Later, from https://blog.qmee.com/online-in-60-seconds-infographic-a-year-later/.
10. Vangie Beal, Insider attack, from https://www.webopedia.com/TERM/I/insider_attack.html.
11. Ian Sutherland, Is Ethical Hacking Actually Ethical or even Legal?, from https://ianhsutherland.com/ethical-hacking/.
12. Is ethical hacking legal?, from https://www.answers.com/Q/Is_ethical_hacking_legal?#slide=2.
13. PCI SSC Data Security Standards Overview, from https://www.pcisecuritystandards.org/pci_security/how.
14. (2010), Payment Card Industry (PCI) Data Security Standard, from https://www.pcisecuritystandards.org/document_library?category=pcidss&document=pci_dss.
15. ISO/IEC 27001:2013, from https://www.iso.org/standard/54534.html.
16. Health Information Privacy, from https://www.hhs.gov/hipaa/index.html.
17. (2002), PUBLIC LAW 107–204—JULY 30, from https://www.sec.gov/answers/about-lawsshtml.html#sox2002.
18. Executive Summary Digital Millennium Copyright Act Section 104 Report, from http://www.copyright.gov/reports/studies/dmca/dmca_executive.html.
19. (1998), The Digital Millennium Copyright Act of 1998 U.S. Copyright Office Summary, from https://www.copyright.gov/legislation/dmca.pdf.
20. (2002), Federal Information Security Management Act (FISMA) Implementation Project, from https://csrc.nist.gov/projects/risk-management.
21. Alan B. Sterneckert, Auditing, from https://www.taylorfrancis.com/books/9781420000047.
22. Joe Jenkins, (2000), Internet Security and Your Business - Knowing the Risks, from https://www.symantec.com/connect/articles/internet-security-and-your-business-knowing-risks.
23. (2006), Critical Infrastructure Threats and Terrorism, from http://www.fas.org/irp/threat/terrorism/sup2.pdf.
24. Juan Andrés Guerrero-Saade, Costin Raiu, Kurt Baumgartner, (2017), Kaspersky Security Bulletin: Threat Predictions for 2018, from https://securelist.com/ksb-threat-predictions-for-2018/83169/.
25. Giovanni Vigna, (2017), The 2018 Cyberthreat Landscape—Predictions and Trends, from https://www.lastline.com/blog/2018-cyberthreat-predictions-and-trends/.
26. Scott Rosenberg, (2017), Firewalls don't stop Hackers. AI might., from https://www.wired.com/story/firewalls-dont-stop-hackers-ai-might/.
27. Nick Ismail, (2017), The role of AI in cyber security, from http://www.information-age.com/role-ai-cyber-security-123465795/.

28. Vangie Beal, Machine Learning, from https://www.webopedia.com/TERM/M/machine-learning.html.
29. ELIEZER KANAL, (2017), Machine Learning in Cybersecurity, from https://insights.sei.cmu.edu/sei_blog/2017/06/machine-learning-in-cybersecurity.html.
30. Machine Learning in Cyber Security Domain – 1: Fundamentals, from https://www.normshield.com/machine-learning-in-cyber-security-domain-1-fundamentals/.
31. Christy Pettey, (2016), Security Embraces Advanced Analytics and Machine Learning, from https://www.gartner.com/smarterwithgartner/security-embraces-advanced-analytics-and-machine-learning/.
32. Kasey Panetta, (2017), Gartner 7 Top Security Predictions for 2017, from https://www.gartner.com/smarterwithgartner/7-top-security-predictions-for-2017/.
33. Stamford, Conn, (2017), Gartner Says Four Vectors Are Transforming the Security Software Market, from https://www.gartner.com/newsroom/id/3731817.
34. (2017), Cybersecurity's Next Frontier: 80+ Companies Using Artificial Intelligence to Secure The Future In One Infographic, from https://www.cbinsights.com/research/cybersecurity-artificial-intelligence-startups-market-map/.
35. (2017), Firewalls Don't Stop Hackers. AI Might, from https://www.wired.com/story/firewalls-dont-stop-hackers-ai-might/.
36. Martin Anderson, (2019), The State of AI in Cyber Security in 2019, from https://www.iflexion.com/blog/ai-in-cyber-security.
37. (2019), Artificial Intelligence in Cybersecurity Market, from https://www.marketsandmarkets.com/Market-Reports/artificial-intelligence-security-market-220634996.html.
38. Remesh Ramachandran, (2019), How Artificial Intelligence Is Changing Cyber Security Landscape and Preventing Cyber Attacks, from https://www.entrepreneur.com/article/339509.
39. TJ Horan, (2018), 5 Keys to Using AI and Machine Learning in Fraud Detection, from https://www.fico.com/blogs/5-keys-using-ai-and-machine-learning-fraud-detection.
40. Chigozie-Okwum C .C. and Ajah Ifeyinwa Angela, (2019), Botnet Identification Using Machine Learning Techniques: A Survey, from https://www.researchgate.net/publication/334284867_Botnet_Identification_Using_Machine_Learning_Techniques_A_Survey.
41. Raghav Bharadwaj, (2019), Artificial Intelligence in Cybersecurity – Current Use-Cases and Capabilities, from https://emerj.com/ai-sector-overviews/artificial-intelligence-cybersecurity/.

Module 02: Footprinting and Reconnaissance

42. Search Operators, from http://www.googleguide.com/advanced_operators_reference.html.
43. Extract Website Information from https://archive.org.
44. Simson Garfinkel and David Cox, (2009), Finding and Archiving the Internet Footprint, from https://simson.net/clips/academic/2009.BL.InternetFootprint.pdf.
45. CHAPTER 2 [Footprinting], from https://e-cq.net/resources/wp/footprinting-encored.pdf.
46. Manic Velocity, Footprinting And The Basics Of Hacking, from http://web.textfiles.com/hacking/footprinting.txt.
47. Regional Internet Registry, from https://en.wikipedia.org/wiki/Regional_Internet_registry.
48. Arik R. Johnson, What is competitive intelligence?, from http://www.aurorawdc.com/whatisci.htm.
49. Manic Velocity, (2002), Footprinting: The Basics of Hacking: Hack in The Box, from https://news.hitb.org/node/5359.
50. P. Mockapetris, (1987), RFC 1034 [Domain Names - Concepts and Facilities], from https://tools.ietf.org/html/rfc1034.
51. Reporting network abuse: Spamming and hacking, from https://www.apnic.net/manage-ip/using-whois/abuse-and-spamming/.
52. Kevin Beaver, How to Use Footprinting to Plan an Ethical Hack, from https://www.dummies.com/programming/networking/how-to-use-footprinting-to-plan-an-ethical-hack/.
53. Ali Jahangiri, Google Hacking, from http://www.alijahangiri.org/publication/Google-Hacking-by-Ali-Jahangiri.pdf.
54. Richard A. Goodman, Competitive Intelligence, from https://www.anderson.ucla.edu/rosenfeld-library/learn/competitive-intelligence#A-1.
55. Sam Bowne, Chapter 1: Footprinting, from http://samsclass.info/124/ppt/ch01.ppt.
56. (1998), Chapter 6: Competitive Intelligence, from http://gaius.cbpp.uaa.alaska.edu/afef/IMIchapter-6.htm.

57. Eddie Sutton, Footprinting: What is it and How Do You Erase Them, from http://www.infosecwriters.com/text_resources/pdf/Footprinting.pdf.
58. Locating Exploits and Finding Targets, from http://media.techtarget.com/searchSoftwareQuality/downloads/Google_Hacking_Penetration_Testers.pdf.
59. Johnny Long, (2008), Google Hacking: Ten security searches that work, from https://searchitchannel.techtarget.com/tip/Google-Hacking-Ten-security-searches-that-work.
60. Google hacking, from https://en.wikipedia.org/wiki/Google_hacking.
61. Google Tips, from http://www.pafumi.net/Google_Tips.html.
62. (2018), Google Hacking Database (GHDB), from https://www.exploit-db.com/google-hacking-database.
63. Nihad A. Hassan, Rami Hijazi, (2018), Open Source Intelligence Methods and Tools_ A Practical Guide to Online Intelligence, Newyork, USA: Apress.
64. (2020), Video search engine, from https://en.wikipedia.org/wiki/Video_search_engine.
65. (2019), Metasearch engine, from https://en.wikipedia.org/wiki/Metasearch_engine#Advantages.
66. Subarna Kumar Das, (2006), Role of Meta Search Engines in Web- Based Information System: Fundamentals and Challenges, from http://ir.inflibnet.ac.in:8080/ir/bitstream/1944/1327/1/445-454.pdf.
67. Rafay Baloch, (2015), Ethical Hacking and Penetration Testing Guide, London, Newyork: CRC Press.
68. Timothy Shim, (2020), How to Access the Dark Web: Browsing Dark Web, TOR Browser, and Onion Websites, from https://www.webhostingsecretrevealed.net/blog/web-tools/tourist-guide-to-dark-web-accessing-the-dark-web-tor-browser-and-onion-websites/.
69. Paul Bischoff, (2018), Step by step guide to safely accessing the dark net and deep web, from https://www.comparitech.com/blog/vpn-privacy/how-to-access-the-deep-web-and-darknet/.
70. (2019), Usenet newsgroup, from https://en.wikipedia.org/wiki/Usenet_newsgroup.
71. (2019), Sherlock – A Tool to Find Usernames Across Social Networks, from https://latesthackingnews.com/2019/01/30/sherlock-a-tool-to-find-usernames-across-social-networks/.
72. Raj Chandel, (2017), Beginner Guide to Footprinting, from https://www.hackingarticles.in/beginner-guide-footprinting/.
73. Raj Chandel, (2017), POST CATEGORY: Footprinting, from https://www.hackingarticles.in/category/footprinting/.
74. Patrick Engebretson, and David Kennedy, (2013), The Basics of Hacking and Penetration Testing, Second Edition, USA: Elsevier.
75. Dafydd Stuttard, and Marcus Pinto, (2011), The Web Application Hacker's Handbook, Second Edition, Indianapolis, Indiana: John Wiley & Sons, Inc.
76. Kadam Parikh, (2017), Email Footprinting- Trace and Email and Collect Information from it..!, from https://libraryofhacks.blogspot.com/2017/06/email-footprinting-trace-email-and.html.

Module 03: Scanning Networks

77. Fyodor, (1998), Remote OS detection via TCP/IP Stack Fingerprinting, from https://nmap.org/nmap-fingerprinting-article.txt.
78. Remote OS Detection, from https://nmap.org/book/osdetect.html.
79. Katherine Knickerbocker, Spoofing, from http://all.net/CID/Attack/papers/Spoofing.html.
80. Explanation of the Three-Way Handshake via TCP/IP, from https://support.microsoft.com/en-us/help/172983/explanation-of-the-three-way-handshake-via-tcp-ip.
81. The Art of Port Scanning - by Fyodor, from https://nmap.org/nmap_doc.html.
82. Hacking Exposed, from https://www.scribd.com/document/62708034/Hacking-Exposed-Book.
83. Chris McNab, Network Security Assessment, from https://www.trustmatta.com/downloads/pdf/Matta_IP_Network_Scanning.pdf.
84. Steven J. Templeton, and Karl E. Levitt, Detecting Spoofed Packets, from http://seclab.cs.ucdavis.edu/papers/DetectingSpoofed-DISCEX.pdf.
85. Antirez, Hping2 Man Page, from http://www.hping.org/manpage.html.
86. Thierry Lagarde, (2014), AutoScan Network, from https://autoscan-network.en.lo4d.com/windows.
87. Avi Kak, (2019), Port Scanning, Vulnerability Scanning, Packet Sniffing, and Intrusion Detection, and Penetration Testing, from https://engineering.purdue.edu/kak/compsec/NewLectures/Lecture23.pdf.

88. Renaud Deraison, Ron Gula, and Todd Hayton, (2009), Passive Vulnerability Scanning Introduction, from https://docs.huihoo.com/nessus/passive_scanning_tenable.pdf.
89. Lance Cottrell, Anonymizer Limitations: Logs, from https://www.livinginternet.com/i/is_anon.htm.
90. D. Dieterle, (2013), SSDP Scanning for UPnP Vulnerabilities, from https://cyberarms.wordpress.com/2013/08/23/ssdp-scanning-for-upnp-vulnerabilities/.
91. Sharan R, Hacking Techniques - Scanning Networks and Countermeasures, from http://hack-o-crack.blogspot.in/2010/12/hacking-techniques-scanning-networks.html.
92. Chris McNab, Network Security Assessment, from https://www.trustmatta.com/downloads/pdf/Matta_IP_Network_Scanning.pdf.
93. Wade Stich, Banner Grabbing with Telnet, from https://siekta.files.wordpress.com/2010/10/bannergrabbing.pdf.
94. Network scanning, from https://searchmidmarketsecurity.techtarget.com/definition/network-scanning.
95. Ryan Spangler, (2003), Analysis of Remote Active Operating System Fingerprinting Tools, from http://www.packetwatch.net/documents/papers/osdetection.pdf.
96. Ryan Spangler, (2003), Analysis of Remote Active Operating System Fingerprinting Tools, from https://securiteam.com/securityreviews/5ZP010UAAI/.
97. (2000), How Anonymizers Work, from https://www.livinginternet.com/i/is_anon_work.htm.
98. (2008), Scanning - HSC Guides - Ethical Hacker, from http://sqlinjections.blogspot.com/2009/04/scanning-hsc-guides-ethical-hacker.html.
99. Lance Spitzner, (2000), Passive Fingerprinting, from https://www.symantec.com/connect/articles/passive-fingerprinting.
100. Prabhaker Mateti, (2001), Port Scanning, from https://web1.cs.wright.edu/~pmateti/Courses/499/Probing/index.html.
101. Wenliang (Kevin) Du, TCP Protocols, from http://www.cis.syr.edu/~wedu/Teaching/cis758/LectureNotes/TCP.doc.
102. (2008), Proxy Servers and Anonymizers, from https://www.gohacking.com/what-is-proxy-server-and-how-it-works/.
103. Port Scanning without the SYN flag, from http://bsd.opennet.ru/base/sec/p49-15.txt.html.
104. Christian Starkjohann, What is sslproxy?, from https://kuix.de/sslproxy/README.txt.
105. Bálint KOZMAN, (2002), Anonymizers, from http://www.fim.uni-linz.ac.at/staff/sonntag/LTAEC_Budapest/Anonymizers/index.html.
106. Firewall/IDS Evasion and Spoofing, from https://nmap.org/book/man-bypass-firewalls-ids.html.
107. (2016), How can the Nmap tool be used to evade a firewall/IDS?, from https://security.stackexchange.com/questions/121900/how-can-the-nmap-tool-be-used-to-evade-a-firewall-ids.
108. (2012), Nmap – Techniques for Avoiding Firewalls, from https://pentestlab.blog/2012/04/02/nmap-techniques-for-avoiding-firewalls/.
109. Rajesh Deodhar, (2011), Advanced Nmap: Scanning Firewalls, from http://opensourceforu.com/2011/02/advanced-nmap-scanning-firewalls/.
110. Taylor Gibb, (2012), Hacker Geek: OS Fingerprinting With TTL and TCP Window Sizes, from https://www.howtogeek.com/104337/hacker-geek-os-fingerprinting-with-ttl-and-tcp-window-sizes/.
111. Evrik Hjelmvik, (2011), Passive OS Fingerprinting, from http://www.netresec.com/?page=Blog&month=2011-11&post=Passive-OS-Fingerprinting.
112. Christoph Eckstein, (2011), OS fingerprinting with IPv6, from https://www.sans.org/reading-room/whitepapers/testing/os-fingerprinting-ipv6-33794.
113. Chris Sanders, (2011), Operating System Fingerprinting with Packets (Part 1), from http://techgenix.com/operating-system-fingerprinting-packets-part1/.
114. Toby Millers, Passive OS Fingerprinting Update, from https://isc.sans.edu/forums/diary/Passive+OS+Fingerprinting+Update/18.
115. (2012), Penetration Testing Lab, from https://pentestlab.blog/2012/08/17/nmap-cheat-sheet/.
116. Host Discovery Techniques, from https://nmap.org/book/host-discovery-techniques.html.
117. Gordon "Fyodor" Lyon, (2008), Nmap_network_scanning, United States: Insecure.Com LLC.
118. Paulino Calderon, (2012), Discovering hosts with UDP ping scans, from https://subscription.packtpub.com/book/networking_and_servers/9781849517485/2/ch02lvl1sec24/discovering-hosts-with-udp-ping-scans.

119. Sairam Jetty, (2018), Network Scanning Cookbook Practical network security using Nmap and Nessus 7, Birmingham, UK: Packt Publishing.
120. (2014), Ping Sweeps and Port Scans, from https://computersecurity123.wordpress.com/2014/07/16/ping-sweeps-and-port-scans/.
121. Kevin Beaver, Prevent Network Hacking with Port Scanners, from https://www.dummies.com/programming/networking/prevent-network-hacking-with-port-scanners/.
122. (2018), When is 'Timestamp' and 'Timestamp Reply' are used in ICMP protocol?, from https://networkengineering.stackexchange.com/questions/50511/when-is-timestamp-and-timestamp-reply-are-used-in-icmp-protocol.
123. (2020), Internet Control Message Protocol, from https://en.wikipedia.org/wiki/Internet_Control_Message_Protocol.
124. Nmap Cheat Sheet, from https://hackt.nl/nmap-cheat-sheet/.
125. Savia Lobo, (2018), Discovering network hosts with ‘TCP SYN’ and ‘TCP ACK’ ping scans in Nmap, from https://hub.packtpub.com/discovering-network-hosts-with-tcp-syn-and-tcp-ack-ping-scans-in-nmaptutorial/.
126. Paulino Calderon, (2012), Discovering hosts with UDP ping scans, from https://subscription.packtpub.com/book/networking_and_servers/9781849517485/2/ch02lvl1sec24/discovering-hosts-with-udp-ping-scans.
127. Port Scanning Techniques, from https://nmap.org/book/man-port-scanning-techniques.html.
128. (2014), Nmap Cheat Sheet, from https://highon.coffee/blog/nmap-cheat-sheet/.
129. TCP FIN, NULL, and Xmas Scans (-sF, -sN, -sX), fromhttps://nmap.org/book/scan-methods-null-fin-xmas-scan.html.
130. TCP Maimon Scan (-sM), from https://nmap.org/book/scan-methods-maimon-scan.html.
131. Philippe Langlois, (2013), SCTPscan Finding entry points to SS7 Networks & Telecommunication Backbones, from https://www.slideshare.net/p1sec/black-hat-europe-philippe-langlois-sctpscan.
132. Nmap Network Scanning IPv6 Scanning (-6), from https://nmap.org/book/port-scanning-ipv6.html.
133. Jason Wood, (2016), Nmap Scanning IPv6 Addresses, from https://www.youtube.com/watch?v=WWHyg4Qjz6c.
134. Nmap Network Scanning Service and Version Detection, from https://nmap.org/book/man-version-detection.html.
135. Nmap Network Scanning Scan Time Reduction Techniques, from https://nmap.org/book/reduce-scantime.html.
136. Taylor Gibb, (2012), Hacker Geek: OS Fingerprinting with TTL and TCP Window Sizes, from https://www.howtogeek.com/104337/hacker-geek-os-fingerprinting-with-ttl-and-tcp-window-sizes/.
137. Erik Hjelmvik, (2011), Passive OS Fingerprinting, from http://www.netresec.com/?page=Blog&month=2011-11&post=Passive-OS-Fingerprinting.
138. Chris Sanders, (2011), Operating System Fingerprinting with Packets (Part 1), from http://techgenix.com/operating-system-fingerprinting-packets-part1/.
139. TCP window size – Wireshark and Windows, from https://microsoftserver2012.wordpress.com/2016/10/06/tcp-window-size-wireshark-and-windows/.
140. Subin Siby, (2019), Default TTL (Time To Live) Values of Different OS, from https://subinsb.com/default-device-ttl-values/.
141. Chris Greer, (2018), How TCP Works - Window Scaling and Calculated Window Size, from https://www.youtube.com/watch?v=2PJVHvthrNU.
142. OS Detection, from https://nmap.org/book/man-os-detection.html.
143. Service and Version Detection, from https://nmap.org/book/man-version-detection.html.
144. Ron Bowes, (2008), My Scripting Experience with Nmap, from https://blog.skullsecurity.org/2008/what-time-is-it .
145. Ron Bowes, File smb-os-discovery, from https://nmap.org/nsedoc/scripts/smb-os-discovery.html .
146. IPv6 fingerprinting, from https://nmap.org/book/osdetect-ipv6-methods.html.
147. Jonathan Hassell, (2003), What is IP spoofing? And 5 ways to prevent it, from https://www.csoonline.com/article/2115848/data-protection-ip-spoofing.html.

Module 04: Enumeration

148. The Dark Side of NTFS (Microsoftâ€™s Scarlet Letter), from http://www.infosecwriters.com/texts.php?op=display&id=53.
149. RPCCLIENT, from http://www.sarata.com/manpages/man1/rpcclient.html.
150. smtp-user-enum User Documentation, from http://pentestmonkey.net/tools/user-enumeration/smtp-user-enum.

151. What is SNMP?, from http://www.wtcs.org/snmp4tpc/snmp.htm.

152. Jan van Oorschot, Jeroen Wortelboer and Dirk Wisse, (2001), SNMP - The Mission Statement, from https://www.symantec.com/connect/articles/windows-2000-snmp-and-security.

153. Joris Evers, (2006), AT&T hack exposes 19,000 identities, from https://www.cnet.com/news/at-38t-hack-exposes-19000-identities/.

154. (2019), Linux finger command, from https://www.computerhope.com/unix/ufinger.htm.

155. (2008), Enumeration, from http://www.vulnerabilityassessment.co.uk/enum.htm.

156. (2009), Net view, from https://docs.microsoft.com/en-us/previous-versions/windows/it-pro/windows-xp/bb490719(v=technet.10)?redirectedfrom=MSDN.

157. Djrand, (2003), Low-Level Enumeration With TCP/IP, from https://packetstormsecurity.com/papers/bypass/Low-Level_Enumeration_With_TCP.txt.

158. LDAP Enumeration, from http://www.vulnerabilityassessment.co.uk/enum_ldap.htm.

159. SMTP Enumeration and more, from http://www.vulnerabilityassessment.co.uk/enum_smtp.htm.

160. (2019), Linux rpcinfo command, from https://www.computerhope.com/unix/urpcinfo.htm.

161. Arun Thomas, (2012), IPsec VPN Penetration Testing with BackTrack Tools, from http://opensourceforu.com/2012/01/ipsec-vpn-penetration-testing-backtrack-tools/.

162. Nadeem Unuth, (2019), What is SIP and How does it work?, from https://www.lifewire.com/what-is-sip-3426669.

163. Irfan Shakeel, (2016), VoIP Network Recon: Footprinting, Scanning, and Enumeration, from http://resources.infosecinstitute.com/voip-network-recon-footprinting-scanning-and-enumeration/#gref.

164. Enumerating and Breaking VoIP, from http://garage4hackers.com/attachment.php?attachmentid=120&d=1321865311.

165. Endler & Collier, (2006), Enumerating a VoIP Network, from http://www.hackingvoip.com/presentations/sample_chapter3_hacking_voip.pdf.

166. (2003), What is RPC?, from https://technet.microsoft.com/en-us/library/cc787851(v=ws.10).aspx.

167. (2003), How RPC work", from https://technet.microsoft.com/en-us/library/cc738291(v=ws.10).aspx.

168. Enumerating Unix RPC Services, from http://etutorials.org/Networking/network+security+assessment/Chapter+12.+Assessing+Unix+RPC+Services/12.1+Enumerating+Unix+RPC+Services/.

169. Adam Bertram, Using Windows 10 Administrative Shares, from https://www.businessnewsdaily.com/11017-windows-10-administrative-shares.html.

170. Sean Wilkins, (2012), TCP/IP Ports and Protocols, from http://www.pearsonitcertification.com/articles/article.aspx?p=1868080.

171. Shahmeer Amir, (2017), Penetration Testing of an FTP Server, from https://shahmeeramir.com/penetration-testing-of-an-ftp-server-19afe538be4b.

172. Raj Chandel, (2017), Penetration Testing on Telnet (Port 23), from https://www.hackingarticles.in/penetration-testing-telnet-port-23/.

173. Network File System (NFS), from https://web.mit.edu/rhel-doc/5/RHEL-5-manual/Deployment_Guide-en-US/ch-nfs.html.

174. (2019), NetBIOS Suffix Definitions, from https://docs.microsoft.com/en-us/openspecs/windows_protocols/ms-brws/0c773bdd-78e2-4d8b-8b3d-b7506849847b.

175. (2019), Enumerate NetBIOS Shares with NBTScan & Nmap Scripting Engine, from https://null-byte.wonderhowto.com/how-to/enumerate-netbios-shares-with-nbtscan-nmap-scripting-engine-0193957/.

176. NET.exe VIEW, from https://ss64.com/nt/net-view.html.

177. (2018), The SNMP Management Information Base (MIB), from https://docs.microsoft.com/en-us/windows/desktop/snmp/the-snmp-management-information-base-mib-.

178. ntpdate(8) - Linux man page, from https://linux.die.net/man/8/ntpdate.

179. Satyam Singh, (2018), Exploiting NFS Share, from https://resources.infosecinstitute.com/exploiting-nfs-share/#gref.

180. Joe Norton, (2017), Exploiting Metasploitable Without Metasploit — NFS Enumeration and Exploiting Misconfiguration, from https://medium.com/@joe_norton/exploiting-metasploitable-without-metasploit-nfs-enumeration-and-exploiting-misconfiguration-86504ccd15b9.

181. (2017), NFS, from https://pentestacademy.wordpress.com/2017/09/20/nfs/.

182. Margaret Rouse, (2019), Network File System (NFS), from https://searchenterprisedesktop.techtarget.com/definition/Network-File-System .
183. (2019), Network File System, from https://en.wikipedia.org/wiki/Network_File_System.
184. (2019), Common SMTP port numbers, from https://docs.mailpoet.com/article/59-default-ports-numbers-smtp-pop-imap.
185. (2012), SMTP User Enumeration, from https://pentestlab.blog/2012/11/20/smtp-user-enumeration/.
186. Duarte Silva, File smtp-enum-users, from https://nmap.org/nsedoc/scripts/smtp-enum-users.html.
187. Heyder Andrade and Nebulus, (2018), SMTP User Enumeration Utility, from https://www.rapid7.com/db/modules/auxiliary/scanner/smtp/smtp_enum.
188. Suzanne Goldlust, (2018), What is DNS Cache snooping?, from https://kb.isc.org/docs/aa-00509.
189. Dejan Lukan, (2012), DNS Cache Snooping, from https://resources.infosecinstitute.com/dns-cache-snooping/#gref.
190. Johannes Weber, (2016), How to walk DNSSEC Zones: dnsrecon, from https://blog.webernetz.net/how-to-walk-dnssec-zones-dnsrecon/.
191. What Are Domain Name System Security Extensions?, from https://whatismyipaddress.com/dnssec.
192. (2014), DNSSEC:NSEC vs. NSEC3, from https://www.internetsociety.org/resources/deploy360/2014/dnssecnsec-vs-nsec3/.
193. DNSSEC – What Is It and Why Is It Important?, from https://www.icann.org/resources/pages/dnssec-what-is-it-why-important-2019-03-05-en.
194. How DNSSEC Works, from https://www.cloudflare.com/dns/dnssec/how-dnssec-works/.
195. NSEC3-Records (Next Secure v. 3), from https://simpledns.com/help/nsec3-records.
196. (2020), Telnet, from https://en.wikipedia.org/wiki/Telnet.
197. Dinesh Thakur, Telnet - What is Telnet?, from http://ecomputernotes.com/computernetworkingnotes/services-and-applications/what-is-telnet.
198. Raj Chandel, (2016), Password Cracking using Nmap, from https://www.hackingarticles.in/password-cracking-using-nmap/.
199. Kevin Beaver, (2014), How to detect and defend against a TCP port 445 exploit and attacks, from https://searchsecurity.techtarget.com/answer/Detecting-and-defending-against-TCP-port-445-attacks.
200. Raj Chandel, (2019), SMB Penetration Testing (Port 445), from https://www.hackingarticles.in/smb-penetration-testing-port-445/.
201. Shahmeer Amir, (2017), Penetration Testing of an FTP Server, from https://shahmeeramir.com/penetration-testing-of-an-ftp-server-19afe538be4b.
202. Raj Chandel, (2017), FTP Penetration Testing on Windows (Port 21), from https://www.hackingarticles.in/ftp-penetration-testing-windows/.
203. Margaret Rouse, (2019), IPv6 (Internet Protocol Version 6), from https://searchnetworking.techtarget.com/definition/IPv6-Internet-Protocol-Version-6.
204. (2020), IPv6, from https://en.wikipedia.org/wiki/IPv6.
205. Srinivas, (2016), SNMP Pentesting, from https://resources.infosecinstitute.com/snmp-pentesting/#gref.
206. (2011), Enumeration Countermeasures, from http://luizfirmino.blogspot.com/2011/09/enumeration-countermeasures.html.
207. Kevin Beaver, (2014), How to detect and defend against a TCP port 445 exploit and attacks, from https://searchsecurity.techtarget.com/answer/Detecting-and-defending-against-TCP-port-445-attacks.
208. Kevin Beaver, Countermeasures against NFC attacks, from https://books.google.co.in/books?id=xuF3Rr2uD3YC&pg=PR13&lpg=PR13&dq=NFS+enumeration+countermeasures&source=bl&ots=S8R5PmXEf7&sig=ACfU3U2rFLwul5j9ztVenCLltdQMwCLd7A&hl=en&sa=X&ved=2ahUKEwiwscLW9JLjAhUQ8HMBHcAUAvYQ6AEwBHoECAkQAQ#v=onepage&q=NFS%20attacks&f=false.
209. Stuart McClure, Joel Scambray, and George Kurtz, (2009), Hacking Exposed 6, Tenth Edition, USA: McGraw-Hill Companies, Inc.
210. Tavis Barr, Nicolai Langfeldt, Seth Vidal, and Tom McNeal, (2002), Security and NFS, from https://www.tldp.org/HOWTO/NFS-HOWTO/security.html#PORTMAPPER-SECURITY.

Module 05: Vulnerability Analysis

211. Microsoft Vulnerability Research (MSVR), from https://www.microsoft.com/en-us/msrc/msvr.
212. Glossary of Vulnerability Testing Terminology, from https://www.ee.oulu.fi/research/ouspg/Glossary/.

213. Thomas R. Peltier, Justin Peltier, and John A. Blackley, (2017), Technical (Bottom-Up) Methodology, from https://www.taylorfrancis.com/books/9780429210747.
214. Renaud Deraison and Ron Gula, (2009), Blended Security Assessments, from https://www.tenable.com/sites/drupal.dmz.tenablesecurity.com/files/uploads/documents/whitepapers/Blended%20Security%20Assesments.pdf.
215. Steven Weil, (2003), How to obtain a high-quality vulnerability assessment, from https://searchsecurity.techtarget.com/tip/How-to-obtain-a-high-quality-vulnerability-assessment.
216. Dr. D. Polemi, G. Valvis, (2005), Vulnerability Assessment Report Format Data Model, from http://www.kti.ae.poznan.pl/conferences/i3e/papers/george_valvis.pdf.
217. (2011), What is a vulnerability assessment?, from http://resecure.me/pdf/17542.pdf .
218. Marcelo Silva, (2012), Vulnerability Assessment, from https://www.slideshare.net/CelloLtd/info-security-vulnerability-assessment.
219. Common Vulnerability Scoring System Calculator, from https://nvd.nist.gov/vuln-metrics/cvss/v3-calculator.
220. (2019), Common Weakness Enumeration, from https://en.wikipedia.org/wiki/Common_Weakness_Enumeration.
221. (2015), Testing Scan Credentials for More Accurate Vulnerability Assessment, from https://www.tripwire.com/state-of-security/vulnerability-management/testing-scan-credentials-for-more-accurate-vulnerability-assessment/.
222. (2011), Credentialed vs Non-Credentialed scans, from https://discussions.qualys.com/thread/10133.
223. Syamini Sreedharan, What is Vulnerability Assessment? Testing Process, VAPT Scan Tool, from https://www.guru99.com/vulnerability-assessment-testing-analysis.html.

Module 06: System Hacking

224. Gregory Kipper, (2003), Steganography in Depth, from https://www.taylorfrancis.com/books/9780429211065.
225. S.J. Templeton, K.E. Levitt, (2003), Detecting spoofed packets, from https://ieeexplore.ieee.org/document/1194882.
226. NTLM Authentication in Java, from http://www.luigidragone.com/software/ntlm-authentication-in-java/.
227. Samir K Bandyopadhyay, Debnath Bhattacharyya, Debashis Ganguly, Swarnendu Mukherjee, and Poulami Das, A Tutorial Review on Steganography, from http://www.jiit.ac.in/jiit/ic3/IC3_2008/IC3-2008/APP2_21.pdf.
228. Ricky Magalhaes, (2003), Using passwords as a defense mechanism to improve Windows security, from http://techgenix.com/passwords_improve_windows_security_part2/.
229. Andreas Westfeld and Andreas Pfitzmann, Attacks on Steganographic Systems, from http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.94.5975&rep=rep1&type=pdf.
230. DaijiSanai and HidenobuSeki, (2004), Optimized Attack for NTLM2 Session Response, from https://www.blackhat.com/presentations/bh-asia-04/bh-jp-04-pdfs/bh-jp-04-seki.pdf.
231. Zhi Wang, Xuxian Jiang, Weidong Cui, and Xinyuan Wang, (2008), Countering Persistent Kernel Rootkits Through Systematic Hook Discovery, from https://www.microsoft.com/en-us/research/publication/countering-persistent-kernel-rootkits-through-systematic-hook-discovery/.
232. Brute force attack - Wikipedia, the free encyclopedia, from https://en.wikipedia.org/wiki/Brute-force_attack.
233. Passwords, from http://media.techtarget.com/searchSecurity/downloads/HackingforDummiesCh07.pdf.
234. The Hack FAQ: Password Basics, from https://www.nmrc.org/pub/faq/hackfaq/hackfaq-04.html.
235. Microsoft Technical Security Notifications, from https://www.microsoft.com/en-us/msrc/technical-security-notifications?rtc=1.
236. Bojan Smojver, (2002), Linux Today - ZDNet Australia: Threats Move Beyond Linux to Windows, from https://www.linuxtoday.com/security/2002121100426scsvnt.
237. Russell Kay, (2006), Sidebar: A Simple Rootkit Example, from https://www.computerworld.com/article/2560499/cybercrime-hacking/sidebar--a-simple-rootkit-example.html.
238. Russell Kay, (2006), Rootkits offer the lure of total control, from https://www.techworld.com/security/rootkits-offer-the-lure-of-total-control-2219/.
239. Steganography (a secretly hidden coding that dates back to ancient ...), from https://wordinfo.info/unit/3403?letter=S&spage=9.
240. Fred B. Schneider, Authentication, from http://www.cs.cornell.edu/Courses/cs513/2000sp/NL10.html.
241. CS513: System Security - Topic Outline, from http://www.cs.cornell.edu/courses/cs513/2005fa/02.outline.html.

242. Mary McMahon, (2014), What Is a Privilege Escalation?, from https://www.wisegeek.com/what-is-a-privilege-escalation.htm.
243. Foon, (2002), Exploiting design flaws in the Win32 API for privilege escalation - Shatter Attacks - How to break Windows, from https://www.helpnetsecurity.com/2002/08/08/exploiting-design-flaws-in-the-win32-api-for-privilege-escalation---shatter-attacks---how-to-break-windows/.
244. Scott Sutherland, (2009), Windows Privilege Escalation Part 1: Local Administrator Privileges, from https://blog.netspi.com/windows-privilege-escalation-part-1-local-administrator-privileges/ .
245. Windows Privilege Escalation Fundamentals, from http://www.fuzzysecurity.com/tutorials/16.html.
246. Siva Ram, (2010), DLL Hijacking Attacks, from https://www.maravis.com/dll-hijacking-attacks/.
247. (2018), Dynamic-Link Library Security, from https://docs.microsoft.com/en-gb/windows/win32/dlls/dynamic-link-library-security?redirectedfrom=MSDN.
248. Srikanth Ramesh, How to Hack Windows Administrator Password, from https://www.gohacking.com/hack-windows-administrator-password./.
249. Aleksandr Matrosov, (2013), Mysterious Avatar rootkit with API, SDK, and Yahoo Groups for C&C communication, from https://www.welivesecurity.com/2013/05/01/mysterious-avatar-rootkit-with-api-sdk-and-yahoo-groups-for-cc-communication/.
250. Win32/Rootkit.Avatar, from https://www.virusradar.com/en/Win32_Rootkit.Avatar/description.
251. James Wyke, The ZeroAccess rootkit, from https://nakedsecurity.sophos.com/zeroaccess2/#Introduction.
252. (2013), Hack Like a Pro: How to Cover Your Tracks & Leave No Trace Behind on the Target System, from https://null-byte.wonderhowto.com/how-to/hack-like-pro-cover-your-tracks-leave-no-trace-behind-target-system-0148123/.
253. Sarah Granger, (2002), The Simplest Security: A Guide To Better Password Practices, from https://www.symantec.com/connect/articles/simplest-security-guide-better-password-practices.
254. Gaining Access Using Application and Operating System Attacks, from https://searchsecurity.techtarget.com/feature/Gaining-access-using-application-and-operating-system-attacks.
255. Jesper M. Johansson, Windows Passwords: Everything You Need To Know, from http://download.microsoft.com/download/a/d/0/ad0f04a3-21b2-4d79-9049-f5fadb632ace/SEC401-JesperJohansson.pdf.
256. Dr-Hack, (2009), Hash injection Attacks in a Windows Network, from https://blog.drhack.net/hash-injection-attacks-in-a-windows-network/.
257. How to prevent Windows from storing a LAN manager hash of your password in Active Directory and local SAM databases, from https://support.microsoft.com/en-us/help/299656/how-to-prevent-windows-from-storing-a-lan-manager-hash-of-your-passwor.
258. System Hacking: Part II, from http://archive.visualstudiomagazine.com/books/chapters/0072260815.pdf.
259. Fred B. Schneider, Authentication, from http://www.cs.cornell.edu/courses/cs513/2000sp/NL10.html.
260. Gary C. Kessler, (2001), Steganography: Hiding Data Within Data, https://www.garykessler.net/library/steganography.html.
261. Gary C. Kessler, (2015), An Overview of Steganography for the Computer Forensics Examiner, from, https://www.garykessler.net/library/fsc_stego.html.
262. Soumyendu Das, Steganography and Steganalysis: Different Approaches, from https://arxiv.org/ftp/arxiv/papers/1111/1111.3758.pdf.
263. Link Local Multicast Name Resolution (LLMNR) and NetBIOS Name Service (NBT-NS), from https://www.cccsecuritycenter.org/remediation/llmnr-nbt-ns.
264. Jon Sternstein, Local Network Attacks: LLMNR and NBT-NS Poisoning, from https://www.sternsecurity.com/blog/local-network-attacks-llmnr-and-nbt-ns-poisoning.
265. LLMNR / NBT-NS Spoofing Attack Network Penetration Testing, from https://www.aptive.co.uk/blog/llmnr-nbt-ns-spoofing/.
266. HollyGraceful, (2015), Stealing Accounts: LLMNR and NBT-NS Spoofing, from https://www.gracefulsecurity.com/stealing-accounts-llmnr-and-nbt-ns-poisoning/.
267. Mucahit Karadag, (2016), What is LLMNR & WPAD and How to Abuse Them During Pentest?, from https://pentest.blog/what-is-llmnr-wpad-and-how-to-abuse-them-during-pentest/.
268. William Hurer-Mackay, (2016), LLMNR and NBT-NS Poisoning Using Responder, from https://www.4armed.com/blog/llmnr-nbtns-poisoning-using-responder/.

269. Link Local Multicast Name Resolution (LLMNR) and NetBIOS Name Service (NBT-NS), from https://www.cccsecuritycenter.org/remediation/llmnr-nbt-ns.

270. (2019), Exploitation for Privilege Escalation, from https://attack.mitre.org/wiki/Technique/T1068.

271. (2018), Dylib Hijacking, from https://attack.mitre.org/wiki/Technique/T1157.

272. Patrick Wardle, (2015), Dylib hijacking on OS X, from https://www.virusbulletin.com/virusbulletin/2015/03/dylib-hijacking-os-x/.

273. (2015), Dylib hijacking on OS X, from https://news.ycombinator.com/item?id=9239411.

274. (2018), Meltdown and Spectre: Critical processor vulnerabilities, from https://www.enisa.europa.eu/publications/info-notes/meltdown-and-spectre-critical-processor-vulnerabilities.

275. Justin Ellingwood, (2018), How To Protect Your Server Against the Meltdown and Spectre Vulnerabilities, from https://www.digitalocean.com/community/tutorials/how-to-protect-your-server-against-the-meltdown-and-spectre-vulnerabilities.

276. Josh Fruhlinger, (2018), Spectre and Meltdown explained: What they are, how they work, what's at risk, from https://www.csoonline.com/article/3247868/vulnerabilities/spectre-and-meltdown-explained-what-they-are-how-they-work-whats-at-risk.html.

277. Tom Ueltschi, Travis Smith, Jared Atkinson, Robby Winchester, (2017), Access Token Manipulation, from https://attack.mitre.org/wiki/Technique/T1134.

278. Access Tokens, from https://msdn.microsoft.com/en-us/library/windows/desktop/aa374909(v=vs.85).aspx.

279. (2017), Application Shimming, from https://attack.mitre.org/wiki/Technique/T1138.

280. Stefan Kanthak and Travis Smith, (2017), File System Permissions Weakness, from https://attack.mitre.org/wiki/Technique/T1044.

281. Travis Smith and Leo Loobeek, (2017), Scheduled Task, from https://attack.mitre.org/wiki/Technique/T1053.

282. (2017), Launch Daemon, from https://attack.mitre.org/wiki/Technique/T1160.

283. (2017), Plist Modification, from https://attack.mitre.org/wiki/Technique/T1150.

284. (2017), Setuid and Setgid, from https://attack.mitre.org/wiki/Technique/T1166.

285. (2017), Web Shell, from https://attack.mitre.org/wiki/Technique/T1100.

286. Agathoklis Prodromou, (2016), An introduction to web Shells-Part-1, from https://www.acunetix.com/blog/articles/introduction-web-shells-part-1/.

287. (2015), Covering Tracks of Attacks, from https://resources.infosecinstitute.com/covering-tracks-of-attacks/#gref.

288. (2018), Microsoft NTLM, from https://docs.microsoft.com/en-us/windows/win32/secauthn/microsoft-ntlm.

289. Amrita Mitra, (2017), What is Pass The Hash Attack?, from https://www.thesecuritybuddy.com/vulnerabilities/what-is-a-pass-the-hash-attack/.

290. (2019), Pass the hash, from https://en.wikipedia.org/wiki/Pass_the_hash.

291. (2007), Pass-The-Hash Toolkit, from https://www.coresecurity.com/corelabs-research-special/open-source-tools/pass-hash-toolkit.

292. Elad Shamir, (2019), Internal Monologue Attack - Retrieving NTLM Hashes without Touching LSASS (Repost), from https://shenaniganslabs.io/2019/01/14/Internal-Monologue.html.

293. (2018), Retrieving NTLM Hashes without touching LSASS: the "Internal Monologue" Attack, from https://www.andreafortuna.org/2018/03/26/retrieving-ntlm-hashes-without-touching-lsass-the-internal-monologue-attack/.

294. Yaron Ziner, (2017), Advanced Techniques Attackers Use to Crack Passwords, from https://resources.infosecinstitute.com/advanced-techniques-attackers-use-crack-passwords/#gref.

295. Adam Chester, (2017), Kerberos AD Attacks - More Roasting with AS-REP, from https://blog.xpnsec.com/kerberos-attacks-part-2/.

296. Jeff Petters, (2018), Kerberos Authentication Explained, from https://www.varonis.com/blog/kerberos-authentication-explained/.

297. Aki Jos, (2018), AS-REP Roasting – Cracking User Account Password, from https://akijosberryblog.wordpress.com/2018/01/17/as-rep-roasting-cracking-user-account-password/.

298. Sean Metcalf, (2015), Cracking Kerberos TGS Tickets Using Kerberoast – Exploiting Kerberos to Compromise the Active Directory Domain, from https://adsecurity.org/?p=2293.

299. Ryan Becwar, and Vincent Le Toux, (2019), Pass the Ticket, from https://attack.mitre.org/techniques/T1097/.
300. Chris Stoneff, (2018), Defending Against Pass-the-Ticket Attacks, from https://www.beyondtrust.com/blog/entry/defending-against-pass-the-ticket-attacks.
301. (2017), Cracking Passwords: 11 Password Attack Methods (And How They Work), from https://datarecovery.com/rd/cracking-passwords-11-password-attack-methods-work/.
302. Jens Steube, (2013), Advanced password guessing, from https://hashcat.net/events/p13/js-apg-htftl20.pdf.
303. William Hurer-Mackay, (2016), How to Perform a Combinator Attack Using Hashcat, from https://www.4armed.com/blog/hashcat-combinator-attack/.
304. Atom, (2010), Automated Password Cracking: Use oclHashcat To Launch A Fingerprint Attack, from https://www.question-defense.com/2010/08/15/automated-password-cracking-use-oclhashcat-to-launch-a-fingerprint-attack.
305. The Different Types of Password Cracking Techniques, from https://password-managers.bestreviews.net/the-different-types-of-password-cracking-techniques/.
306. Daniel Turner, (2012), Hashcat Per Position Markov Chains, from https://www.trustwave.com/en-us/resources/blogs/spiderlabs-blog/hashcat-per-position-markov-chains/.
307. Lisa Bock, Defend against password attacks, from https://www.linkedin.com/learning/ethical-hacking-system-hacking/defend-against-password-attacks.
308. Daniel Doc Sewell, Offline Password Cracking: The Attack and the Best Defense, from https://www.alpinesecurity.com/blog/offline-password-cracking-the-attack-and-the-best-defense-against-it.
309. Samantha Rorke, (2017), Protecting your Network against Brute Force Password attacks, from https://www.lookingglasscyber.com/blog/threat-intelligence-insights/protecting-network-brute-force-password-attacks/.
310. Adam Gordon, (2017), How to prevent password attacks and other exploits, from https://searchsecurity.techtarget.com/tip/How-to-prevent-password-attacks-and-other-exploits.
311. What is a buffer overflow? Learn about buffer overrun vulnerabilities, exploits & attacks, from https://www.veracode.com/security/buffer-overflow.
312. Buffer Overflow Attack with Example, from https://www.geeksforgeeks.org/buffer-overflow-attack-with-example/.
313. (2020), Buffer overflow, from https://en.wikipedia.org/wiki/Buffer_overflow.
314. Differences between Stack and Heap, from http://net-informations.com/faq/net/stack-heap.htm.
315. Stack vs Heap Memory Allocation, from https://www.geeksforgeeks.org/stack-vs-heap-memory-allocation/.
316. Subhashini Rai, (2019), What Is Stack Based Buffer Overflow?, from https://hackersonlineclub.com/stack-based-buffer-overflow/.
317. Heap overflow and Stack overflow, from https://www.geeksforgeeks.org/heap-overflow-stack-overflow/.
318. (2018), 32-Bit Windows Buffer Overflows Made Easy, from https://veteransec.com/2018/09/10/32-bit-windows-buffer-overflows-made-easy/.
319. Tomasz Andrzej Nidecki, (2019), What Is a Buffer Overflow, from https://www.acunetix.com/blog/web-security-zone/what-is-buffer-overflow/.
320. Buffer Overflow Attack, from https://www.imperva.com/learn/application-security/buffer-overflow/.
321. Avoiding Buffer Overflows and Underflows, from https://developer.apple.com/library/archive/documentation/Security/Conceptual/SecureCodingGuide/Articles/BufferOverflows.html.
322. Buffer Overflow Defenses, from http://wwwusers.di.uniroma1.it/~parisi/Risorse/BO-defense.pdf.
323. Bradley Green, Exploitation Development and Implementation, from http://www.iup.edu/WorkArea/DownloadAsset.aspx?id=170683.
324. Mutaz Alsallal, (2018), Identifying Named Pipe Impersonation and Other Malicious Privilege Escalation Techniques, from https://securityintelligence.com/identifying-named-pipe-impersonation-and-other-malicious-privilege-escalation-techniques/.
325. Azeria, Privilege Escalation, from https://azeria-labs.com/privilege-escalation/.
326. (2017), Unquoted Service Path, from https://pentestlab.blog/2017/03/09/unquoted-service-path/.
327. Jonathan, (2015), Common Windows Privilege Escalation Vectors, from https://www.toshellandback.com/2015/11/24/ms-priv-esc/.

328. Abed, (2017), Pivots and Relays for Extreme Post-Exploitation Control, from https://www.semurity.com/pivots-and-relays-for-extreme-post-exploitation-control/.

329. (2012), Post Exploitation – Port Forwarding, from https://pentestlab.blog/2012/04/22/post-exploitation-port-forwarding/.

330. (2018), Access Tokens, from https://docs.microsoft.com/en-us/windows/win32/secauthz/access-tokens?redirectedfrom=MSDN.

331. Windows Shim Database (SDB) Parser (shims), from https://tzworks.net/prototype_page.php?proto_id=33.

332. (2020), sudo, from https://en.wikipedia.org/wiki/Sudo.

333. Margaret Rouse, (2005), sudo (superuser do), from https://searchsecurity.techtarget.com/definition/sudo-superuser-do.

334. Privilege Escalation, from https://chryzsh.gitbooks.io/pentestbook/privilege_escalation_-_linux.html.

335. Raj Chandel, (2018), Linux Privilege Escalation using SUID Binaries, from https://www.hackingarticles.in/linux-privilege-escalation-using-suid-binaries/.

336. Barrow, (2016), Use a Misconfigured SUID Bit to Escalate Privileges & Get Root, from https://null-byte.wonderhowto.com/how-to/use-misconfigured-suid-bit-escalate-privileges-get-root-0173929/.

337. David Lodge, (2015), Exploiting SUID Executables, from https://www.pentestpartners.com/security-blog/exploiting-suid-executables/.

338. Justin Ellingwood, (2018), How To Protect Your Server Against the Meltdown and Spectre Vulnerabilities, from https://www.digitalocean.com/community/tutorials/how-to-protect-your-server-against-the-meltdown-and-spectre-vulnerabilities.

339. (2018), Meltdown and Spectre Side-Channel Vulnerability Guidance, from https://www.us-cert.gov/ncas/alerts/TA18-004A.

340. Mike Burns, (2018), Running the Intel Meltdown Detection Tool, from https://medium.com/think-stack/running-the-intel-meltdown-detection-tool-d4d92735e606.

341. James Sanders, (2019), Spectre and Meltdown explained: A comprehensive guide for professionals, from https://www.techrepublic.com/article/spectre-and-meltdown-explained-a-comprehensive-guide-for-professionals/.

342. (2019), Execution, from https://attack.mitre.org/tactics/TA0002/.

343. (2020), Keystroke logging, from https://en.wikipedia.org/wiki/Keystroke_logging.

344. (2020), Defending Against Keyloggers, from https://documentation.logmein.com/webhelp/EN/SecDoc/LogMeIn/c_common_Security_keylogger.html.

345. Linda McGlasson, (2010), How to Beat Keyloggers, from https://www.bankinfosecurity.com/how-to-beat-keyloggers-a-2999.

346. (2018), How to Protect Yourself From Keyloggers, from https://www.expressvpn.com/internet-privacy/guides/protect-yourself-from-keyloggers-monitoring/.

347. (2013), Defending against Keyloggers, from https://homeguntraining.wordpress.com/2013/04/23/defendingagainstkeylogger/.

348. Ben Alonzo, (2017), Simple tips for defending against keylogger attacks, from https://ultratechlife.com/tech/simple-tips-for-defending-against-keylogger-attack/.

349. Five ways to keep keyloggers away from your data, from https://www.nbins.com/blog/cyber-risk/five-ways-to-keep-keyloggers-away/.

350. (2020), Rootkit, from https://en.wikipedia.org/wiki/Rootkit.

351. Rootkit, from https://www.imperva.com/learn/application-security/rootkit/.

352. (2018), LoJax: First UEFI rootkit found in the wild, courtesy of the Sednit group, from https://www.welivesecurity.com/wp-content/uploads/2018/09/ESET-LoJax.pdf.

353. (2018), LoJax UEFI Rootkit Used in Cyberespionage, from https://www.trendmicro.com/vinfo/us/security/news/cyber-attacks/lojax-uefi-rootkit-used-in-cyberespionage.

354. Andrei Raul Ardelean, Claudiu Ștefan Coblis, Cristofor Ochinca, and Cristian Alexandru Istrate, Inside Scranos – A Cross Platform, Rootkit-Enabled Spyware Operation, from https://www.bitdefender.com/files/News/CaseStudies/study/253/Bitdefender-Whitepaper-RootKit-CREAT3432-en-EN.pdf.

355. Catalin Cimpanu, (2019), Scranos rootkit expands operations from China to the rest of the world, from https://www.zdnet.com/article/scranos-rootkit-expands-operations-from-china-to-the-rest-of-the-world/.

356. Garrett Gross, (2016), Rootkit Detection: Techniques and Best Practices, from https://www.alienvault.com/blogs/security-essentials/rootkit-detection-techniques-and-best-practices.

357. (2020), Rootkit Detection, from https://en.wikipedia.org/wiki/Rootkit#Detection.

358. Randy Franklin Smith, (2006), Defending Against Rootkits, from https://www.itprotoday.com/security/defending-against-rootkits.

359. Mike Chapple, (2015), Security Matters: Rootkit Attacks and How to Prevent Them, from http://www.gocertify.com/articles/security-matter-rootkit-attacks-and-how-to-prevent-them.html.

360. Jesus Olguin, (2016), Steganalysis, the Counterpart of Steganography, from https://www.trustwave.com/en-us/resources/blogs/spiderlabs-blog/steganalysis-the-counterpart-of-steganography/.

361. Clear-EventLog, from https://docs.microsoft.com/en-us/powershell/module/microsoft.powershell.management/clear-eventlog?view=powershell-5.1.

362. Ivan Jenic, (2018), How to clear the Event Log in Windows 10, 7, from https://windowsreport.com/clear-event-log-windows-8/.

363. (2020), Atomic Test #1 - Clear Logs, from https://github.com/redcanaryco/atomic-red-team/blob/master/atomics/T1070/T1070.md#atomic-test-1---clear-logs.

364. Anand Khanse, (2018), Cipher command line tool in Windows 10, from https://www.thewindowsclub.com/cipher-command-line-tool-windows.

365. (2018), MS-DOS and Windows command line cipher command, from https://www.computerhope.com/cipher.htm.

366. Nihad Ahmad Hassan, Rami Hijazi, and Helvi Salminen, (2017), Data Hiding Techniques in Windows OS: A Practical Approach to Investigation and Defense, from https://books.google.co.in/books?id=sy2lCgAAQBAJ&pg=PA267&lpg=PA267&dq=techniques+for+wiping+attack+traces&source=bl&ots=__ihU2l6i6&sig=ACfU3U2zKfyKvtl6BE4MBS3iSLqroM8x1w&hl=en&sa=X&ved=2ahUKEwic2JSI9pzjAhVTdCsKHVEEAaA4ChDoATANegQICRAB#v=onepage&q=cipher.exe&f=false.

367. Trisha, (2014), How to Disable Virtual Memory in Windows 10, from https://www.trishtech.com/2014/11/disable-virtual-memory-in-windows-10/.

368. (2017), Enable Last Access timestamp for files and folder in Windows, from https://www.opentechguides.com/how-to/article/windows-10/129/enable-last-access-time.html.

369. Richard Falk, (2016), How to disable Sleep Mode or Hibernation, from https://www.pugetsystems.com/labs/support-software/How-to-disable-Sleep-Mode-or-Hibernation-793/.

370. Clear Thumbnails Cache – Guide for Windows XP, Vista, 7, 8, 8.1, 10, from https://neosmart.net/wiki/clear-thumbnails-cache/.

Module 07: Malware Threats

371. The corporate threat posed by email Trojans, from https://www.gfisoftware.de/whitepapers/network-protection-against-trojans.pdf

372. Fausi Qattan & Fredrik Thernelius, (2004), Master's Thesis, from http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.112.1869&rep=rep1&type=pdf.

373. Commodon Communications - Threats to your Security on the Internet, from http://www.commodon.com/threat/index.htm.

374. Van Hauser/THC, Placing Backdoors Through Firewalls, from https://www.cgisecurity.com/lib/placing_backdoors_through_firewalls.txt.

375. David Wells, (1996), Wrappers, from http://www.objs.com/survey/wrap.htm.

376. Trojans FAQ, from http://techgenix.com/trojans-faq/.

377. How to block ICMP tunneling?, from https://listserv.icsalabs.com/pipermail/firewall-wizards/1999-July/006060.html.

378. Candid Wueest, (2015), Financial Trojans in 2014: Takedowns contributed to 53 percent drop in infections, but threat is still prevalent, from https://www.symantec.com/connect/blogs/financial-trojans-2014-takedowns-contributed-53-percent-drop-infections-threat-still-prevalent.

379. Candid Wueest, (2015), The state of financial Trojans 2014, from https://www.symantec.com/content/dam/symantec/docs/white-papers/state-of-financial-trojans-2014-en.pdf.

380. (2013), Battling with Cyber Warriors- Exploit Kits, from https://resources.infosecinstitute.com/battling-cyber-warriors-exploit-kits/.

381. Joshua Cannell, (2013), Tools of the Trade: Exploit Kits, from https://blog.malwarebytes.com/cybercrime/2013/02/tools-of-the-trade-exploit-kits/.

382. Pierluigi Paganini, (2014), New private Exploit-Kit "Infinity" on the underground, from http://securityaffairs.co/wordpress/25013/cyber-crime/new-private-exploit-kit-infinity-available-underground.html.
383. Marco Preuss, (2013), ChewBacca - a New Episode of Tor-based Malware, from https://securelist.com/chewbacca-a-new-episode-of-tor-based-malware/58192/.
384. Yotam Gottesman, (2014), RSA Uncovers New POS Malware Operation Stealing Payment Card & Personal Information, from https://community.rsa.com/community/products/netwitness/blog/2014/01/30/rsa-uncovers-new-pos-malware-operation-stealing-payment-card-personal-information.
385. Pierluigi Paganini, (2012), Skynet, the potential use of Tor as a bulletproof botnet, from http://securityaffairs.co/wordpress/10960/cyber-crime/skynet-the-potential-use-of-tor-as-a-bulletproof-botnet.html.
386. (2013), SpyGate RAT, from http://spygate-rat.blogspot.in/.
387. Marshall Brain, How Computer Viruses Work, from http://www.mindpride.net/root/Extras/how-stuff-works/how_computer_viruses_work.htm.
388. Virus Protection, from http://www.mindpride.net/root/services/virus_alert_map_advisory.htm.
389. Paul Boutin, (2003), An inside view of the worm that crashed the Internet in 15 minutes, from https://www.wired.com/2003/07/slammer/.
390. Mark Russinovich, (2019), Autoruns for Windows v13.96, from https://docs.microsoft.com/en-us/sysinternals/downloads/autoruns.
391. Mark Russinovich, (2011), TCPView v3.05, from https://docs.microsoft.com/en-us/sysinternals/downloads/tcpview.
392. Norman Book on Computer Viruses, from http://download.norman.no/manuals/eng/BOOKON.PDF.
393. Carey Nachenberg, Understanding and Managing Polymorphic Viruses, from https://www.symantec.com/avcenter/reference/striker.pdf.
394. Dr. Alan Solomon and Robert M. Slade, 1990 - VX BBS & Little Black Book (AT&T Attack), 1991 - Tequila, 2001 - Gnuman, Winux Windows/Linux Virus, 2004 - Trojan.Xombe, Randex, Bizex, Witty, from https://www.cknow.com/cms/vtutor/virus-history-summary.html.
395. Ransomware, from https://en.wikipedia.org/wiki/Ransomware.
396. ANGLER EK AND ANOTHER CRYPTOWALL SAMPLE, from http://malware-traffic-analysis.net/2014/05/28/index.html.
397. GENERAL REMOVAL INSTRUCTIONS, from https://www.f-secure.com/v-descs/guides/general-removal-instructions.shtml.
398. Andreinick05's Batch Virus Maker, from http://www.nerdprogrammer.in/2014/03/andreinick05s-batch-virus-maker.html.
399. DEADLY VIRUS MAKER, from www.nerdprogrammer.in/2014/02/deadlines-virus-maker.html.
400. Praneeth, (2012), DeadLine's Virus Maker 1.8.5, from http://internetfalcon.blogspot.in/2012/11/deadlines-virus-maker-185.html.
401. Computer Worms, from https://userpages.umbc.edu/~dgorin1/432/worms.htm.
402. Kaoru Hayashi, (2014), IoT Worm Used to Mine Cryptocurrency, from https://www.symantec.com/connect/blogs/iot-worm-used-mine-cryptocurrency.
403. Kaoru Hayashi, (2013), Linux Worm Targeting Hidden Devices, from https://www.symantec.com/connect/blogs/linux-worm-targeting-hidden-devices.
404. (2014), Exploit Kits, from https://www.trustwave.com/Resources/SpiderLabs-Blog/Terror-Exploit-Kit--More-like-Error-Exploit-Kit/?page=1&year=0&month=0&tag=Exploit+Kits&LangType=1033.
405. Worm, from https://www.trendmicro.com/vinfo/us/threat-encyclopedia/malware/worm.
406. R. A. Hettinga, (2003), Random Scanning Worms and Sapphire/Slammer's PRNG..., from https://www.mail-archive.com/cryptography@wasabisystems.com/msg03503.html.
407. King Adnan Anjum, (2010), Reverse WWW Shell - Covert Channels Using HTTP, from https://hackguide4u.blogspot.com/2010/03/reverse-www-shell-covert-channels-using.html.
408. DaBoss, (2013), File Extensions, from https://www.cknow.com/cms/vtutor/file-extensions.html.
409. DaBoss, (2013), Companion Files, from https://www.cknow.com/cms/vtutor/companion-files.html.
410. Ed Skoudis, (2003), Trojan horses, from http://www.informit.com/articles/article.aspx?p=102181&seqNum=2.
411. Shahram Monshi Pouri, Nikunj Modi, Trojans and Backdoors, from https://www.it.uu.se/edu/course/homepage/sakdat/ht06/assignments/pm/programme/modi-monshipouri.pdf.
412. (2016), What Is A Banking Trojan And How Does It Work, from https://thecentexitguy.com/what-is-a-banking-trojan-and-how-does-it-work/.

413. Candid Wueest, (2017), ISTR Financial Threats Review 2017, from https://www.symantec.com/content/dam/symantec/docs/security-center/white-papers/istr-financial-threats-review-2017-en.pdf.

414. Swati Khandelwal, (2017), New Windows Trojan Spreads MIRAI Malware To Hack More IoT Devices, from https://thehackernews.com/2017/02/mirai-iot-botnet-windows.html#author-info.

415. Catalin Cimpanu, (2017), New Malware Intentionally Bricks IoT Devices, from https://www.bleepingcomputer.com/news/security/new-malware-intentionally-bricks-iot-devices/.

416. Farzad, (2010), Introduction to Trojans and Backdoors, from https://www.symantec.com/connect/articles/introduction-trojans-and-backdoors.

417. (2017), Virus hoax, from https://www.sophos.com/en-us/threat-center/threat-analyses/hoaxes/virus-hoax.aspx.

418. Emilian Varsanov, (2017), Zeus Virus Scam (2017 Alert Hoax) , from https://www.virusresearch.org/zeus-virus-scam-2017-alert-hoax/.

419. (2016), Kaspersky Security Bulletin 2016. Story of the year The Ransomware revolution, from https://media.kaspersky.com/en/business-security/kaspersky-story-of-the-year-ransomware-revolution.pdf.

420. (2016), Locky Ransom Virus Removal and Preventive Vaccination. Everything you need to Know, from https://www.techsupportall.com/beware-locky-ransomware-need-know-locky/.

421. Tomas Meskauskas, (2017), Locky Ransomware [Updated], from https://www.pcrisk.com/removal-guides/9807-locky-ransomware.

422. Andra Zaharia, (2016), Security Alert: New Locky Ransomware Shows Off through Rampant Distribution, from https://heimdalsecurity.com/blog/security-alert-new-locky-ransomware-shows-off-rampant-distribution/.

423. (2017), What you need to know about the WannaCry Ransomware, from https://www.symantec.com/blogs/threat-intelligence/wannacry-ransomware-attack.

424. Russell Brandom, (2017), Petya ransomware authors demand $250,000 in first public statement since the attack, from https://www.theverge.com/2017/7/5/15922216/petya-notpetya-ransomware-authors-bitcoin-demand-decrypt.

425. What is the FAT Virus?, from https://www.wisegeek.com/what-is-the-fat-virus.htm#comments.

426. Rohil Dubey, Security and Malware - Web Scripting Viruses, from http://touque.ca/EC/students/DubeyR/ics_minor_project.html.

427. Margaret Rouse, (2007), E-Mail Virus, from https://searchmidmarketsecurity.techtarget.com/definition/email-virus.

428. E-Mail Virus, from https://www.techopedia.com/definition/15802/email-virus.

429. Common Trojan Ports, from https://www.pcsecurityworld.com/75/common-trojan-ports.html.

430. (2019), Commonly Used Port, from https://attack.mitre.org/techniques/T1043/.

431. (2016), Necurs Botnet Returns With Updated Locky Ransomware In Tow, from https://www.proofpoint.com/us/threat-insight/post/necurs-botnet-returns-with-updated-locky-ransomware-in-tow.

432. Luis Mendieta, (2018), DreamBot Campaign Dreams Big, from https://www.anomali.com/blog/dreambot-campaign-dreams-big.

433. Gurubaran S, (2017), Dreambot Banking Trojan Delivered via Resume-Themed Email, from https://gbhackers.com/dreambot-banking-trojan/.

434. (2019), Point-of-sale malware, from https://en.wikipedia.org/wiki/Point-of-sale_malware.

435. (2013), Point-of-Sale Malware Threats, from https://www.secureworks.com/research/point-of-sale-malware-threats.

436. Sergey Yunakovsky, (2017), Neutrino modification for POS-terminals, from https://securelist.com/neutrino-modification-for-pos-terminals/78839/.

437. (2016), Point of Sale (POS), from https://blog.malwarebytes.com/threats/point-of-sale-pos/.

438. (2017), New Trojan Attacks Point-Of-Sale Systems Seeking Card Info, from https://www.cyberianit.com/2017/07/26/new-trojan-attacks-point-of-sale-systems-seeking-card-info/.

439. (2019), BasBanke: Trend-setting Brazilian banking Trojan, from https://securelist.com/basbanke-trend-setting-brazilian-banking-trojan/90365/.

440. Yiyi Miao, (2017), Most Destructive Malware of All Time, from https://www.opswat.com/blog/most-destructive-malware-all-time.

441. Ben Herzog, (2018), Godzilla Loader and the Long Tail of Malware, from https://research.checkpoint.com/godzilla-loader-and-the-long-tail-of-malware/.

442. (2020), Alert (TA18-201A) Emotet Malware, from https://www.us-cert.gov/ncas/alerts/TA18-201A.
443. (2020), Trojan.Downloader, from https://blog.malwarebytes.com/detections/trojan-downloader/.
444. Norah Armstrong, Inside-Out Attacks. Covert Channel Attacks Inside-out Attacks, from https://docplayer.net/864961-Inside-out-attacks-ivan-buetler-csnc-ch-covert-channel-attacks-inside-out-attacks-seite-1-glarnischstrasse-7-postfach-1671-ch-8640-rapperswil.html.
445. (2015), RIG Exploit Kit Source Code Leak - The End or Just the Beginning of RIG?, from https://www.trustwave.com/en-us/resources/blogs/spiderlabs-blog/rig-exploit-kit-source-code-leak-the-end-or-just-the-beginning-of-rig/.
446. David Maciejak and Kenny Yongjian Yang, (2018), Dharma Ransomware: What It's Teaching Us, from https://www.fortinet.com/blog/threat-research/dharma-ransomware--what-it-s-teaching-us.html.
447. Jake Doevan, (2020), Dharma ransomware. 34 Variants listed. 2019 removal instructions, from https://www.2-spyware.com/remove-dharma-ransomware-virus.html.
448. Lena Fuks, (2019), 10 Ransomware Attacks You Should Know About in 2019, from https://www.allot.com/blog/10-ransomware-attacks-2019/.
449. Ransomware, from https://www.trendmicro.com/vinfo/us/security/news/ransomware/page/1.
450. Allan Liska, (2019), 4 Ransomware Trends to Watch in 2019, from https://www.recordedfuture.com/ransomware-trends-2019/.
451. Nat BotPak, (2019), eCh0raix ransomware: Targets Linux file storage servers, from https://en.secnews.gr/185636/ech0raix-ransomware-sto-stochastro-diakomistes-apothikefsis-archeion-linux/.
452. (2019), eCh0raix Ransomware Found Targeting QNAP Network-Attached Storage Devices, from https://www.trendmicro.com/vinfo/au/security/news/cybercrime-and-digital-threats/ech0raix-ransomware-found-targeting-qnap-network-attached-storage-devices.
453. Gurubaran S, (2019), New eCh0raix Ransomware Attacking Linux File Storage Servers, from https://gbhackers.com/qnap-ech0raix-ransomware/.
454. Sergiu Gatlan, (2019), New eCh0raix Ransomware Brute-Forces QNAP NAS Devices, from https://www.bleepingcomputer.com/news/security/new-ech0raix-ransomware-brute-forces-qnap-nas-devices/.
455. Tomas Meskauskas, (2018), Samsam Ransomware, from https://www.pcrisk.com/removal-guides/9992-samsam-ransomware.
456. Dorka Palotay and Peter Mackenzie, (2018), SamSam Ransomware Chooses Its Targets Carefully, from https://www.sophos.com/en-us/medialibrary/PDFs/technical-papers/SamSam-ransomware-chooses-Its-targets-carefully-wpna.pdf.
457. Penny Hoelscher, (2019), What are virus hoaxes, from https://www.comparitech.com/antivirus/virus-hoax/.
458. Ventsislav Krastev, (2018), Google Critical Security Alert Virus Scam (Gmail) – How to Remove (2019), from https://sensorstechforum.com/google-critical-security-alert-virus-scam-gmail-remove/.
459. Moseley, (2019), Two – thirds of all antivirus applications in Android are fraud, from https://cybersguards.com/two-thirds-of-all-antivirus-applications-in-android-are-fraud/.
460. Catalin Cimpanu, (2019), Two-thirds of all Android antivirus apps are frauds, from https://www.zdnet.com/article/two-thirds-of-all-android-antivirus-apps-are-frauds/.
461. (2019), Fileless threats, from https://docs.microsoft.com/en-us/windows/security/threat-protection/intelligence/fileless-threats.
462. Mary Branscombe, (2019), What is fileless malware and how do you protect against it?, from https://www.techrepublic.com/article/what-is-fileless-malware-and-how-do-you-protect-against-it/.
463. Kate Brew, (2019), Fileless Malware Detection: A Crash Course, from https://cybersecurity.att.com/blogs/security-essentials/fileless-malware-detection.
464. Simon Floreza, Donald Castillo, and Mark Manahan, (2018), Security 101: Defending Against Fileless Malware, from https://www.trendmicro.com/vinfo/in/security/news/security-technology/security-101-defending-against-fileless-malware#documentexploits.
465. Lenny Zeltser, (2018), How Fileless Malware Infections Start, from https://blog.minerva-labs.com/how-fileless-malware-infections-start.
466. Jareth, (2017), Fileless malware: Invisible threat or scaremongering hype?, from https://blog.emsisoft.com/en/29070/fileless-malware-attacks/.
467. What Is Fileless Malware?, from https://www.mcafee.com/enterprise/en-in/security-awareness/ransomware/what-is-fileless-malware.html.

468. Fileless Malware Attacks, from https://d3pakblog.wordpress.com/2018/05/05/d34n6_fileless-malware-attacks-intro/.
469. Pedro Tavares, (2018), The Art of Fileless Malware, from https://resources.infosecinstitute.com/art-fileless-malware/#gref.
470. Edmund Brumaghin, (2019), Divergent: "Fileless" NodeJS Malware Burrows Deep Within the Host, from https://blog.talosintelligence.com/2019/09/divergent-analysis.html.
471. Manohar Ghule and Mohd Sadique, (2019), Fileless malware campaign roundup, from https://www.zscaler.com/blogs/research/fileless-malware-campaign-roundup.
472. (2019), Vaporworm Threat: What You Need to Know About the Next Evolution of Malware, from https://www.avertium.com/vaporworm-threat-what-you-need-to-know-about-the-next-evolution-of-malware/.
473. Sudais Asif, (2019), Thousands of Windows PCs infected by Nodersok/Divergent fileless malware, from https://www.hackread.com/windows-pcs-infected-nodersok-divergent-fileless-malware/.
474. Dor Zvi, (2019), Obfuscated Fileless Malware in Cyberattackers' Toolkits: A Closer Look, from https://www.mimecast.com/blog/2019/06/obfuscated-fileless-malware-in-cyberattackers-toolkits-a-closer-look/.
475. Xiaopeng Zhang, (2019), Analysis of a Fresh Variant of the Emotet Malware, from https://www.fortinet.com/blog/threat-research/analysis-of-a-fresh-variant-of-the-emotet-malware.html.
476. Emotet, from https://www.malwarebytes.com/emotet/.
477. Max Heinemeyer, (2019), Glimpsing inside the trojan horse: An insider analysis of Emotet, from https://www.darktrace.com/en/blog/glimpsing-inside-the-trojan-horse-an-insider-analysis-of-emotet/.
478. Vishal Thakur, (2018), Malware analysis: decoding Emotet, part 1, from https://blog.malwarebytes.com/threat-analysis/2018/05/malware-analysis-decoding-emotet-part-1/.
479. (2018), Emotet Trojan, from https://www.cyberswachhtakendra.gov.in/alerts/emotet.html.
480. (2018), SamSam Ransomware Campaigns, from https://www.secureworks.com/research/samsam-ransomware-campaigns.
481. (2018), SamSam ransomware: controlled distribution for an elusive malware, from https://blog.malwarebytes.com/threat-analysis/2018/06/samsam-ransomware-controlled-distribution/.
482. Andrea Lelli, (2019), Dismantling a fileless campaign: Microsoft Defender ATP's Antivirus exposes Astaroth attack, from https://www.microsoft.com/security/blog/2019/07/08/dismantling-a-fileless-campaign-microsoft-defender-atp-next-gen-protection-exposes-astaroth-attack/.
483. Adam Mansour, (2017), How to Block Fileless Malware, from https://www.intelligonetworks.com/blog/fileless-malware.
484. David Strom, (2019), How to Defend Your Organization Against Fileless Malware Attacks, from https://securityintelligence.com/how-to-defend-your-organization-against-fileless-malware-attacks/.
485. (2018), Fileless Malware: What It Is and How to Stop It, from https://www.tripwire.com/state-of-security/security-awareness/fileless-malware-stop/.
486. Sharron Malaver, (2018), How to Protect Against Fileless Malware Attacks, from https://blog.minerva-labs.com/how-to-protect-against-fileless-malware-attacks.
487. Margaret Rouse, (2019), Fileless malware attack, from https://whatis.techtarget.com/definition/fileless-infection-fileless-malware.
488. Stephen Cooper, (2018), Fileless malware attacks explained, from https://www.comparitech.com/blog/information-security/fileless-malware-attacks/.
489. (2018), Fileless Malware the Stealth Attacker, from https://www.allot.com/resources/TB_FILELESS_MALWARE_THREAT_BULLETIN.pd_.pdf.

Module 08: Sniffing

490. Kyle Lai, (2002), Change MAC Address on Win2K & XP, from https://seclists.org/pen-test/2002/Nov/25.
491. Christopher R. Russel, (2001), Penetration Testing with dsniff, from http://www.ouah.org/dsniffintr.htm
492. Telephone tapping or wiretapping, from https://en.wikipedia.org/wiki/Telephone_tapping.
493. Dashdorj Yamkhin, and Youjip Won, Modeling and Analysis of Wireless LAN Traffic, from http://www.dmclab.hanyang.ac.kr/files/publication/journals/international/200911_08.pdf.
494. Sakun, (2011), Overview of Layer 2 Switched Networks and Communication, from http://www.sakunsharma.in/2011/07/overview-layer-2-switched-networks-communication/.
495. R. Droms, (1997), Dynamic Host Configuration Protocol, from https://www.ietf.org/rfc/rfc2131.txt.
496. Yusuf Bhaiji, Understanding, Preventing, Defending Against Layer 2 Attacks, from https://www.sanog.org/resources/sanog15/sanog15-yusuf-l2-security.pdf.

497. Satya P Kumar Somayajula, Yella. Mahendra Reddy, and Hemanth Kuppili, (2011), A New Scheme to Check ARP Spoofing: Prevention of MAN-IN-THE-MIDDLE Attack, from http://www.ijcsit.com/docs/Volume%202/vol2issue4/ijcsit2011020420.pdf.

498. Yusuf Bhaiji, LAYER 2 ATTACKS & MITIGATION TECHNIQUES, from https://www.sanog.org/resources/sanog7/yusuf-L2-attack-mitigation.pdf.

499. Adam Barth, Juan Caballero, and Dawn Song, Secure content sniffing for Web browsers or How to stop papers from reviewing themselves, from https://www.adambarth.com/papers/2009/barth-caballero-song.pdf.

500. Undetectable sniffing on Ethernet, from https://www.askapache.com/hacking/sniffing-ethernet-undetected/.

501. Kirk Hausman, Diane Barrett, and Martin Weiss, (2003),Identifying Nonessential Services and Attacks > Attacks, from http://www.informit.com/articles/article.asp?p=98121&seqNum=2.

502. ARP cache poisoning /ARP spoofing, from https://su2.info/doc/arpspoof.php.

503. Address Resolution Protocol (ARP), from www.erg.abdn.ac.uk/users/gorry/course/inet-pages/arp.html.

504. Packages, from https://www.debian.org/distrib/packages.

505. The Hacker's Ethic, from http://web.textfiles.com/ezines/HWA/hwa-hn34.txt.

506. Alberto Ornaghi and Marco Valleri, Man in the middle attacks, from https://www.blackhat.com/presentations/bh-europe-03/bh-europe-03-valleri.pdf.

507. Tom Olzak, (2006), DNS Cache Poisoning: Definition and Prevention, from https://adventuresinsecurity.com/Papers/DNS_Cache_Poisoning.pdf.

508. Daiji Sanai, (2001), Detection of Promiscuous Nodes using ARP packets, from http://www.securityfriday.com/promiscuous_detection_01.pdf.

509. (2018), Catalyst 4500 Series Switch Cisco IOS Software Configuration Guide, 12.2(25)EW, from https://www.cisco.com/c/en/us/td/docs/switches/lan/catalyst4500/12-2/25ew/configuration/guide/conf/port_sec.html.

510. Raj Chandel, (2017), DHCP Penetration Testing, from https://www.hackingarticles.in/dhcp-penetration-testing/.

511. (2016), 7 Popular Layer 2 Attacks, from http://www.pearsonitcertification.com/articles/article.aspx?p=2491767.

512. (2019), VLAN hopping, from https://en.wikipedia.org/wiki/VLAN_hopping#Switch_spoofing.

513. VLAN Hopping, from https://networklessons.com/cisco/ccnp-switch/vlan-hopping.

514. Pam, (2018), VLAN Hopping: How to Prevent an Attack, from https://www.alienvault.com/blogs/security-essentials/vlan-hopping-and-mitigation.

515. (2018), Common Attack Types on Switches, from https://digitalfortresslk.wordpress.com/2018/03/22/common-attack-types-on-switches/.

516. Valter Popeskic, STP Layer 2 attack – Manipulating Spanning Tree Protocol settings, from https://howdoesinternetwork.com/2012/stp-attack.

517. (2019), Protecting Against an STP Attack, from https://www.ccexpert.us/configuration-mode/protecting-against-an-stp-attack.html.

518. (2011), BPDU Guard, BPDU Filter, Root Guard, Loop Guard & UDLD, from http://ericleahy.com/index.php/bpdu-guard-bpdu-filter-root-guard-loop-guard-udld/.

519. What is BPDU Guard and how to configure BPDU Guard in Cisco Switches, from http://www.omnisecu.com/ccna-security/what-is-bpdu-guard-and-how-to-configure-bpdu-guard-in-cisco-switches.php.

520. (2007), Configuring Spanning Tree PortFast, BPDU Guard, BPDU Filter, UplinkFast, BackboneFast, and Loop Guard, from https://www.cisco.com/c/en/us/td/docs/switches/lan/catalyst4000/8-2glx/configuration/guide/stp_enha.html.

521. Jay Miah, (2017), Loop Guard Concept and Implementation, from http://www.jay-miah.co.uk/loop-guard-concept-and-implementation/.

522. (2018), Catalyst 6500 Release 12.2SY Software Configuration Guide, from https://www.cisco.com/c/en/us/td/docs/switches/lan/catalyst6500/ios/12-2SY/configuration/guide/sy_swcg/spanning_tree_features.html#27629.

523. (2019), Cisco Nexus 5000 Series NX-OS Software Configuration Guide, from https://www.cisco.com/c/en/us/td/docs/switches/datacenter/nexus5000/sw/configuration/guide/cli/CLIConfigurationGuide/BasicEthernet.html.

Module 09: Social Engineering

524. Margaret Rouse, (2011), Spear Phishing, from http://searchsecurity.techtarget.com/definition/spear-phishing.

525. Terry Turner, Social Engineering – Can Organizations Win the Battle?, from http://www.infosecwriters.com/text_resources/pdf/Social_Engineering_Can_Organizations_Win.pdf.
526. Sharon Gaudin, Social Engineering: The Human Side Of Hacking, from http://www.crime-research.org/library/Sharon2.htm.
527. Anti-Phishing Resources, from https://www.antiphishing.org/resources/.
528. (2007), Phishing and bogus emails: HM Revenue and Customs examples, from https://www.gov.uk/government/publications/phishing-and-bogus-emails-hm-revenue-and-customs-examples.
529. (2014), How to Protect Insiders from Social Engineering Threats, from https://docs.microsoft.com/en-us/previous-versions/tn-archive/cc875841(v=technet.10)?redirectedfrom=MSDN.
530. (2014), Security Threats, from https://docs.microsoft.com/en-us/previous-versions/tn-archive/cc723507(v=technet.10)?redirectedfrom=MSDN.
531. Melissa Guenther, (2001), Social Engineering, from http://www.iwar.org.uk/comsec/resources/sa-tools/Social-Engineering.pdf.
532. Bodek Frak, (2008), Gone Phishing, from http://web4.uwindsor.ca/units/its/insight/insight.nsf/babe0ebac149fc7b852567d700715d2a/d1b60efa750dfe21852573d7004cbbd8!OpenDocument.
533. Phishing: Examples & its prevention methods, from http://chowkamleeng.blogspot.com/2008/06/phishing-examples-its-prevention.html.
534. Gunter Ollmann, The Phishing Guide (Part 1), from http://www.technicalinfo.net/papers/Phishing.html.
535. Rachna Dhamija, (2006), Why phishing works, from https://dl.acm.org/doi/10.1145/1124772.1124861.
536. (2009), Social engineering, from https://searchsecurity.techtarget.com/definition/social-engineering.
537. The Social Engineering Framework, from https://www.social-engineer.org/framework/attack-vectors/impersonation/.
538. Smishing, vishing, and phishing… oh my!, from https://www.forensicaccountingservices.com/fraudvault/smishing-vishing-and-phishing/.
539. (2017), Ponemon Institute's 2017 Cost of Data Breach Study: Global Overview, from https://www-01.ibm.com/common/ssi/cgi-bin/ssialias?htmlfid=SEL03130WWEN&.
540. Jake Stroup, (2017), The Many Types of Identity Theft, from https://www.thebalance.com/the-8-types-of-identity-theft-1947176.
541. Clari Melo, (2014), Get to Know These Common Types of ID Theft, from https://www.igrad.com/articles/8-types-of-identity-theft.
542. (2015), The 10 Major Types of Identity Theft, from https://www.idtheftauthority.com/types-of-identity-theft/.
543. (2011), The 6 Types of Identity Theft, from https://securingtomorrow.mcafee.com/consumer/family-safety/the-6-types-of-identity-theft/.
544. (2015), Identity Theft, from https://completeid.com/types-of-identity-theft/.
545. (2015), Social engineering techniques: Pretexting, diversion theft, phishing, from https://sgros-students.blogspot.com/2015/11/pretexting-diversion-theft-phishing.html.
546. (2020), Social engineering (security), from https://en.wikipedia.org/wiki/Social_engineering_(security)#Other_types.
547. Kevin Mitnick, What is social engineering?, from https://www.knowbe4.com/what-is-social-engineering/#1.
548. Chris Brunau, (2019), 5 Types of Social Engineering Attacks, from https://www.datto.com/blog/5-types-of-social-engineering-attacks.
549. Pierluigi Paganini, (2019), The Most Common Social Engineering Attacks, from https://resources.infosecinstitute.com/common-social-engineering-attacks/#gref.
550. The Social Engineering Framework, from https://www.social-engineer.org/framework/influencing-others/pretexting/successful-pretexting/.
551. Erdal Ozkaya and Yuri Diogenes, Cybersecurity - Attack and Defense Strategies, from https://learning.oreilly.com/library/view/cybersecurity-attack/9781788475297/6a6d16cf-64bb-411e-bba2-ecbd10ad2d88.xhtml.
552. 2020 Cost of Insider Threats Global Report, from https://www.observeit.com/cost-of-insider-threats/.
553. (2018), 2018 Cost of Insider Threats: Global, from https://153j3ttjub71nfe89mc7r5gb-wpengine.netdna-ssl.com/wp-content/uploads/2018/04/ObserveIT-Insider-Threat-Global-Report-FINAL.pdf.

554. George Moraetes, (2017), The CISO's Guide to Managing Insider Threats, from https://securityintelligence.com/the-cisos-guide-to-managing-insider-threats/.
555. (2016), Managing Insider Threat, from https://www.ey.com/Publication/vwLUAssets/EY-managing-inside-threat/$FILE/EY-managing-inside-threat.pdf.
556. Facts + Statistics: Identity theft and cybercrime, from https://www.iii.org/fact-statistic/facts-statistics-identity-theft-and-cybercrime.
557. Linda Musthaler, (2008), 13 best practices for preventing and detecting insider threats, from https://www.networkworld.com/article/2280365/13-best-practices-for-preventing-and-detecting-insider-threats.html.
558. Insider Threat Prevention Best Practices, from https://www.netwrix.com/Insider_Threat_Prevention_Best_Practices.html.

Module 10: Denial-of-Service

559. (2006), Denial of Service Attacks: Teardrop and Landÿÿ, from http://users.tkk.fi/~lhuovine/study/hacker98/dos.html.
560. (2006), CERT warns of networked denial of service attacks – Computerworld, from http://www.computerworld.com/action/pages.do?command=viewPage&pagePath=/404.
561. Stephen M. Specht and Ruby B. Lee, (2004), Distributed Denial of Service: Taxonomies of Attacks, Tools and Countermeasures, from http://palms.ee.princeton.edu/PALMSopen/DDoS%20Final%20PDCS%20Paper.pdf.
562. Craig A. Huegen, (2005), Denial of Service Attacks: "Smurfing", from http://www.pentics.net/denial-of-service/white-papers/smurf.cgi.
563. Remotely Triggered Black Hole Filtering in IP Version 6 for Cisco IOS, Cisco IOS XE, and Cisco IOS XR Software, from https://www.cisco.com/c/en/us/about/security-center/ipv6-remotely-triggered-black-hole.html.
564. Frank Kargl, Jörn Maier, Stefan Schlott, and Michael Weber, Protecting Web Servers from Distributed Denial of Service Attacks, from http://www10.org/cdrom/papers/409/.
565. (1997), Denial of Service Attacks, from https://www.cert.org/information-for/denial_of_service.cfm?.
566. Denial of service, from https://searchsecurity.techtarget.com/definition/denial-of-service.
567. Vladimir Golubev, (2005), DoS attacks: crime without penalty, from http://www.crime-research.org/articles/1049/.
568. Gunter Ollmann, (2009), Botnet Communication Topologies, from http://www.technicalinfo.net/papers/PDF/WP_Botnet_Communications_Primer_(2009-06-04).pdf.
569. Renaud BIDOU, Fighting the Botnet Ecosystem, from http://www.iv2-technologies.com/FightingBotnetEcosystem.pdf.
570. Ping of death, from https://searchsecurity.techtarget.com/definition/ping-of-death.
571. Jason Anderson, (2001), An Analysis of Fragmentation Attacks, from http://www.ouah.org/fragma.html.
572. Mariusz Burdach, (2003), Hardening the TCP/IP stack to SYN attacks, from https://www.symantec.com/connect/articles/hardening-tcpip-stack-syn-attacks.
573. Deepak Singh Rana, Naveen Garg, and Sushil Kumar Chamoli, (2012), Citations: TCP SYN Flooding and IP Spoofing Attacks (ResearchIndex), from http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.434.8352&rep=rep1&type=pdf.
574. Stephen Specht and Ruby Lee, (2003), Taxonomies of Distributed Denial of Service Networks, Attacks Tools, and Countermeasures, from https://www.princeton.edu/~rblee/DDoS%20Survey%20Paper_v7final.doc.
575. Gary C. Kessler, (2000), Defenses against Distributed Denial-Of-Service, from https://www.garykessler.net/library/ddos.html.
576. The Distributed Reflection DoS Attack, from https://www.grc.com/sn/sn-008.pdf.
577. Steve Gibson, (2002), Distributed Reflection Denial of Service Bandwidth Consumption, from https://homes.cs.washington.edu/~arvind/cs425/doc/drdos.pdf.
578. Haining Wang, Danlu Zhang, and Kang G. Shin, (2002), SYN Attack, from https://ieeexplore.ieee.org/document/1019404/.
579. Aaron Sullivan, 2001, An Audit of Active Directory Security, from https://www.symantec.com/connect/articles/audit-active-directory-security-part-2.
580. Denial of Service, from https://www.mycert.org.my/en/services/advisories/mycert/2017/main/detail/1277/index.html.
581. (2000), Denial of Service Attack in NetBIOS Services, from https://www.kb.cert.org/vuls/id/32650.
582. Wireless DoS, from https://www.cisco.com/c/en/us/td/docs/wireless/technology/wips/deployment/guide/WiPS_deployment_guide.html#pgfId-43390

583. Abhishek Singh, (2005), Demystifying Denial-Of-Service attacks, part one, from https://www.symantec.com/connect/articles/demystifying-denial-service-attacks-part-one.

584. Denial-of-service attack, from https://en.wikipedia.org/wiki/Denial-of-service_attack.

585. Kevin Poulsen, (2010), New: Cyberattack Against WikiLeaks Was Weak, from https://www.wired.com/2010/11/wikileaks-attack/.

586. What is a DDoS Attack, from https://www.digitalattackmap.com/understanding-ddos/.

587. Glenn Carl and George Kesidis, (2009), Denial-of-Service Attack-Detection Techniques, https://www.evernote.com/shard/s9/note/b11a8c31-8651-4d74-acf9-1fb1b3c0f090/wishi/crazylazy#st=p&n=b11a8c31-8651-4d74-acf9-1fb1b3c0f090.

588. Distributed denial-of-service attack, from http://searchsecurity.techtarget.com/sDefinition/0,,sid14_gci557336,00.html.

589. Taxonomies of Distributed Denial of Service Networks, Attacks, Tools, and Countermeasures, from http://www.princeton.edu/~rblee/DDoS%20Survey%20Paper_v7final.doc.

590. Glenn Carl, (2006), Denial-of-Service Attack-Detection Techniques, from https://www.computer.org/csdl/mags/ic/2006/01/w1082-abs.html.

591. Stephen M. Specht and Ruby B. Lee, (2003), Distributed Denial of Service: Taxonomies of Attacks, Tools and Countermeasures, from http://palms.ee.princeton.edu/PALMSopen/DDoS%20Final%20PDCS%20Paper.pdf.

592. Vijay C Uyyuru, Prateek Arora, and Terry Griffin, Denial of Service (DoS), from http://www.cse.unt.edu/~6581s001/vijay_dos1.ppt.

593. School Safety and Security with the Cisco SAFE Security Architecture, from https://www.cisco.com/c/en/us/td/docs/solutions/Enterprise/Security/safek12.html.

594. Denial of Service (DoS), Distributed DoS (DDoS), from http://www.atlguide2000.com/security/index.php?act=view&aid=193.

595. (2007), Denial Of services [botnet] (DoS), from http://www.go4expert.com/forums/showthread.php?t=3184.

596. SYN Flood Attack, from https://www.cloudflare.com/learning/ddos/syn-flood-ddos-attack/.

597. (2009), Analysis of Fragmented Packet Traffic, from https://www.caida.org/research/traffic-analysis/fragments/.

598. Zobair Khan, (2015), Basics on DDos, from https://www.slideshare.net/kzobair/ddosbdnog.

599. Brian Prince, (2013), Multi-vector DDoS Attacks Grow in Sophistication, from http://www.securityweek.com/multi-vector-ddos-attacks-grow.

600. Dancho Danchev, (2009), Malware-infected WinRAR distributed through Google AdWords, from http://www.zdnet.com/article/malware-infected-winrar-distributed-through-google-adwords/.

601. Angad Singh, (2017), A hacker hijacked Chrome Extension to push Malware, from https://www.officialhacker.com/chrome-extension-to-push-malware/.

602. Stelian Pilci, (2015), Remove vikingwebscanner.com virus (Fake AdwCleaner), from https://malwaretips.com/blogs/vikingwebscanner-com-removal/.

603. (2016), Mobile Security Threats on The Rise as Hackers Can Launch DDoS Attacks on Their Mobile Phones, from https://security.radware.com/ddos-threats-attacks/cyber-attacks-in-the-palm-of-your-hand/.

604. Aishwarya Sreekanth, Prashant Sri, and Teemu Vartiainen, (2016), Dyn DDOS Cyberattack – a case study, from https://webcache.googleusercontent.com/search?q=cache:w8CBTmpDAMcJ:https://mycourses.aalto.fi/pluginfile.php/457047/mod_folder/content/0/Cyber%2520Ghosts.pdf%3Fforcedownload%3D1+&cd=1&hl=en&ct=clnk&gl=in.

605. Scott Hilton, (2016), Dyn Analysis Summary of Friday October 21 Attack, from https://dyn.com/blog/dyn-analysis-summary-of-friday-october-21-attack/.

606. Gurubaran S, (2018), Pulse Wave Heavy DDoS Attack to Take Down Multiple Protected Target Networks, from https://gbhackers.com/new-ddos-attack-pulse-wave/.

607. Zero-day DDoS Attack (0day DDoS attack), from https://ddos-guard.net/en/terminology/attack_type/zero-day-ddos-attack-0day-ddos-attack.

608. 35 Types of DDoS Attacks Explained, from https://javapipe.com/blog/ddos-types/.

609. UDP Flood Attack, from https://www.cloudflare.com/learning/ddos/udp-flood-ddos-attack/.

610. UDP Based Amplification Attacks, from https://instasafe.com/udp-based-amplification-attacks/.

611. Sam Kottler, (2018), DDoS Incident Report, from https://github.blog/2018-03-01-ddos-incident-report/.

612. Lily Hay Newman, (2018), GitHub Survived the Biggest DDoS Attack Ever Recorded, from https://www.wired.com/story/github-ddos-memcached/.
613. (2020), Rate Limiting, from https://en.wikipedia.org/wiki/Rate_limiting.

Module 11: Session Hijacking

614. (2006), hunt(1) - Linux man page, from https://linux.die.net/man/1/hunt.
615. (2006), Web Application Attacks – Intro, from www.netprotect.ch/downloads/webguide.pdf.
616. Steps in Session Hijacking, from https://www.hackguide4u.com/2010/03/steps-in-session-hijacking.html.
617. Session Hijacking, from https://www.imperva.com/learn/application-security/session-hijacking/.
618. Adnan Anjum, Spoofing Vs Hijacking, from https://www.hackguide4u.com/2010/03/spoofing-vs-hijacking.html.
619. Lee Lawson, (2005), Session Hijacking Packet Analysis, from https://www.scribd.com/document/53979390/3479.
620. Session hijacking attack, from https://owasp.org/www-community/attacks/Session_hijacking_attack.
621. Shray Kapoor, Session Hijacking Exploiting TCP, UDP and HTTP Sessions, from http://www.infosecwriters.com/text_resources/pdf/SKapoor_SessionHijacking.pdf.
622. David Endler, (2001), Brute-Force Exploitation of Web Application Session IDs, from https://www.cgisecurity.com/lib/SessionIDs.pdf.
623. Robert Auger, Credential and Session Prediction, from http://projects.webappsec.org/w/page/13246918/Credential%20and%20Session%20Prediction.
624. Trojan horse, from https://searchsecurity.techtarget.com/definition/Trojan-horse.
625. (2008), Prevention from Session Hijacking, from http://hydtechie.blogspot.com/2008/08/prevention-from-session-hijacking.html.
626. Harsh Kevadia, (2013), Session Hijacking, from https://www.slideshare.net/harshjk/session-hijacking-by-harsh-kevadiya.
627. (2009), Man-in-the-middle attack, from https://owasp.org/www-community/attacks/Man-in-the-middle_attack.
628. (2009), Man-in-the-browser attack, from https://owasp.org/www-community/attacks/Man-in-the-browser_attack.
629. Session Hijacking: A Primer, from http://www.cs.binghamton.edu/~steflik/cs455/sessionhijacking.htm.
630. (2009), Session fixation, from https://www.owasp.org/index.php?title=Session_fixation&setlang=es.
631. Session Hijacking in Windows Networks, from http://www.sans.org/reading_room/whitepapers/windows/session_hijacking_in_windows_networks_2124.
632. (2001), CERT® Advisory CA-2001-09 Statistical Weaknesses in TCP/IP Initial Sequence Numbers, from http://www.cert.org/advisories/CA-2001-09.html.
633. (2009), Connection Security and IPsec, from https://docs.microsoft.com/en-us/previous-versions/windows/it-pro/windows-server-2008-R2-and-2008/cc771593(v=ws.10).
634. Laurent Joncheray, (1995), A Simple Active Attack Against TCP, from http://www.blyx.com/public/docs/security/tcp_attack.pdf.
635. (2009), IPSec Architecture, from https://docs.microsoft.com/en-us/previous-versions//bb726946(v=technet.10)?redirectedfrom=MSDN.
636. Mohit Kumar, (2012), CRIME: New SSL/TLS attack for Hijacking HTTPS Sessions, from https://thehackernews.com/2012/09/crime-new-ssltls-attack-for-hijacking.html.
637. Dennis Fisher, (2012), Crime Attack Uses Compression Ratio of TLS Requests as Side Channel to Hijack Secure Sessions, from https://threatpost.com/crime-attack-uses-compression-ratio-tls-requests-side-channel-hijack-secure-sessions-091312/77006/.
638. (2013), BEAST vs. CRIME Attack, from https://resources.infosecinstitute.com/beast-vs-crime-attack/#gref.
639. Amrita Mitra, (2017), What is the CRIME Attack?, from https://www.thesecuritybuddy.com/vulnerabilities/what-is-crime-attack/.
640. Agathoklis Prodromou, (2019), TLS/SSL Explained – Examples of a TLS Vulnerability and Attack, Final Part, from https://www.acunetix.com/blog/articles/tls-vulnerabilities-attacks-final-part/.
641. Nick Lewis, How can an HTTPS session get hijacked with the Forbidden attack?, from https://searchsecurity.techtarget.com/answer/How-can-an-HTTPS-session-get-hijacked-with-the-Forbidden-attack.

642. Dan Goodin, (2016), “Forbidden attack” makes dozens of HTTPS Visa sites vulnerable to tampering, from https://arstechnica.com/information-technology/2016/05/faulty-https-settings-leave-dozens-of-visa-sites-vulnerable-to-forgery-attacks/.
643. Mark Wycislik-Wilson, (2016), Decade-old 'forbidden attack' vulnerability affects HTTPS Visa sites, from https://betanews.com/2016/05/26/https-forbidden-attack/.
644. Pierluigi Paganini, (2016), Dozens of VISA HTTPS-protected sites vulnerable to Forbidden attack, from http://securityaffairs.co/wordpress/47724/breaking-news/forbidden-attack.html.
645. (2017), What Is HSTS and How Do I Implement It?, from https://www.globalsign.com/en/blog/what-is-hsts-and-how-do-i-use-it/.
646. HTTP Strict Transport Security, from https://en.wikipedia.org/wiki/HTTP_Strict_Transport_Security.
647. (2017), HTTP Strict Transport Security Cheat Sheet, from https://owasp.org/www-project-cheat-sheets/cheatsheets/HTTP_Strict_Transport_Security_Cheat_Sheet.html.
648. Justin Johnson, (2016), What is HSTS?, from https://blog.stackpath.com/glossary/hsts/.
649. Sjoerd Langkemper, (2017), Prevent session hijacking with token binding, from https://www.sjoerdlangkemper.nl/2017/07/05/prevent-session-hijacking-with-token-binding/.
650. (2019), HTTP Public Key Pinning (HPKP), from https://developer.mozilla.org/en-US/docs/Web/HTTP/Public_Key_Pinning.
651. HTTP Public Key Pinning, from https://en.wikipedia.org/wiki/HTTP_Public_Key_Pinning.
652. Bjørn Johansen, (2017), What is a Session Donation Attack?, from https://bjornjohansen.no/wordpress-session-donation-attack.
653. Alek Amrani, Session Donation, from https://www.defcon.org/images/defcon-17/dc-17-presentations/defcon-17-alek_amrani-session_donation.pdf.
654. Ramesh Lingappa, (2018), What is Session Hijacking and How You can Stop it, from https://www.freecodecamp.org/news/session-hijacking-and-how-to-stop-it-711e3683d1ac/.
655. Man-in-the-Middle (MITM) Attacks, from https://www.rapid7.com/fundamentals/man-in-the-middle-attacks/.
656. (2019), Man-in-the-Middle Attacks and How to Avoid them, from https://www.netsparker.com/blog/web-security/man-in-the-middle-attack-how-avoid/.
657. Bojana Dobran, (2019), What are Man in the Middle Attacks & How to Prevent MITM Attack with Examples, from https://phoenixnap.com/blog/man-in-the-middle-attacks-prevention.

Module 12: Evading IDS, Firewalls, and Honeypots

658. (2006), Measuring Security Threats with Honeypot Technology, from http://www.honeynet.org/papers/individual/sane-2004.pdf.
659. (2006), SecurityFocus: Honeytokens -The Other Honeypot, from https://www.symantec.com/connect/articles/honeytokens-other-honeypot.
660. (2006), Hardware Firewalls, from http://cybercoyote.org/security/hardware.shtml.
661. (2006), Intrusion Detection System (IDS) Evasion, from http://complianceandprivacy.com/WhitePapers/iDefense-IDS-Evasion/iDefense_IDSEvasion_20060510.pdf.
662. Intrusion detection system evasion techniques, from https://en.wikipedia.org/wiki/Intrusion_detection_system_evasion_techniques.
663. Tarek S.Sobh, (2005), Wired and wireless intrusion detection system: Classifications, good characteristics and state-of-the-art, from https://www.sciencedirect.com/science/article/pii/S092054890500098X.
664. (1998), Defeating Sniffers and Intrusion Detection Systems, from http://www.phrack.org/issues.html?issue=54&id=10.
665. Adam Gowdiak, (2003), Techniques used for bypassing firewall systems, from https://www.terena.org/activities/tf-csirt/meeting9/gowdiak-bypassing-firewalls.pdf.
666. IT Infrastructure Security Plan, from https://www.sciencedirect.com/science/article/pii/B9781597490887500098.
667. What is a firewall?, from https://kb.iu.edu/data/aoru.html.
668. Wassim El-Hajj, Hazem Hajj, Zouheir Trabelsi, and Fadi Aloul, (2010), Updating snort with a customized controller to thwart port scanning, from http://www.aloul.net/Papers/faloul_scn10.pdf.
669. Satyajit, What is HoneyPot?, from https://www.securityhunk.in/2010/06/what-is-honeypot.html.
670. Ashley Poland, (2017), How to Set Up a Honey Pot, from http://www.ehow.com/how_5245821_set-up-honey-pot.html.

671. Martin Roesch, Writing Snort Rules, from https://paginas.fe.up.pt/~mgi98020/pgr/writing_snort_rules.htm.
672. Intrusion detection system evasion techniques, from https://en.wikipedia.org/w/index.php?title=Intrusion_detection_system_evasion_techniques&oldid=311670246.
673. Sumit Siddharth, (2005), Evading NIDS, revisited, from https://www.symantec.com/connect/articles/evading-nids-revisited.
674. How To Access Blocked / Bypass Blocked Websites, from http://www.computingunleashed.com/how-to-access-blocked.html.
675. How do I use a Proxy Server?, from https://whatismyipaddress.com/using-proxies.
676. Steve G. Belovich, Firewall Fairytales, from http://www.iqmtm.com/PDF_presentations/IQ_Firewall_Fairytales_June2010-1.pdf.
677. Vern Paxson and Mark Handley, Defending Against Network IDS Evasion, from http://www.raid-symposium.org/raid99/PAPERS/Paxson.pdf.
678. (1996), Phrack Magazine Volume Seven, Issue Forty-Nine File 06 of ..., from http://phrack.org/issues/49/6.html.
679. Frank Wiles, Quick-Tip: SSH Tunneling Made Easy, from https://www.revsys.com/writings/quicktips/ssh-tunnel.html.
680. Firewall Basics, from http://www.unixgeeks.org/security/newbie/security/firewall.html.
681. Tony Bradley, Free Intrusion Detection (IDS) and Prevention (IPS) Software, from https://www.lifewire.com/ids-and-prevention-ips-software-2487316.
682. Bypassing Firewalls, from https://flylib.com/books/en/3.500.1.95/1/.
683. Vangie Beal, Intrusion detection system, from https://www.webopedia.com/TERM/I/intrusion_detection_system.html.
684. IDS: Re: Polymorphic Shellcode detection, from https://seclists.org/focus-ids/2003/May/22.
685. Circuit-Level Gateway, from http://www.softheap.com/internet/circuit-level-gateway.html.
686. Vicomsoft Firewall Q&A, from http://www.vicomsoft.com/knowledge/reference/firewalls1.html.
687. J. Christian Smith, (2000), Covert Shells - Introduction, from http://www.gray-world.net/papers/covertshells.txt.
688. Peter Kieltyka, ICMP Shell, from http://icmpshell.sourceforge.net/.
689. Mark Burnett, (2001), Running Snort on IIS Web Servers Part 2: Advanced Techniques, from http://www.securityfocus.com/infocus/1316.
690. Mark Burnett, (2001), Running Snort on IIS Web Servers Part 2: Advanced Techniques, from https://www.symantec.com/connect/articles/running-snort-part-2.
691. Niels Provos, (2003), A Virtual Honeypot Framework, from http://www.citi.umich.edu/techreports/reports/citi-tr-03-1.pdf.
692. Brittany Day, (2000), Network Intrusion Detection Using Snort, from https://linuxsecurity.com/features/features/network-intrusion-detection-using-snort.
693. Paul Innella and Oba McMillan, (2001), An Introduction to IDS, from https://www.symantec.com/connect/articles/introduction-ids.
694. Ricky M. Magalhaes, (2003), Host-Based IDS vs Network-Based IDS, from http://techgenix.com/hids_vs_nids_part1/.
695. Paul Innella, (2001), The Evolution of Intrusion Detection Systems, from https://www.symantec.com/connect/articles/evolution-intrusion-detection-systems.
696. ntsecurity.nu - ack tunneling, from http://ntsecurity.nu/papers/acktunneling/.
697. SecuriTeam™ - ACK Tunneling Trojans, from https://securiteam.com/securityreviews/5OP0P156AE/.
698. Mike, Firewalk, from www.blackhat.com/presentations/bh-usa-99/Route/bh-us-99-schiffman.ppt.
699. Huynh Phi Long, Chapter 06: Network Security Using Cisco IOS IPS (Part02), from http://ciscodocuments.blogspot.in/2011/05/chapter-06-network-security-using-cisco_25.html.
700. About firewalls, from https://kb.iu.edu/d/aoru.
701. Pei Guo, How to bypass a firewall, from https://cosec.bit.uni-bonn.de/fileadmin/user_upload/teaching/06ws/06ws-computer-security/49692243.pdf.
702. Frank Wiles, Quick-Tip: SSH Tunneling made easy, from https://www.revsys.com/writings/quicktips/ssh-tunnel.html.
703. Reto E. Haeni, (1997), Firewall Penetration Testing, from http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.42.1117&rep=rep1&type=pdf.
704. Thomas H. Ptacek and Timothy N. Newsham, (1998), Insertion, Evasion and Denial of Service: Eluding Network Intrusion Detection, from https://insecure.org/stf/secnet_ids/secnet_ids.pdf.

705. David D. Rude II and Jayson Jean, Intrusion Detection System (IDS) Evasion, from http://complianceandprivacy.com/WhitePapers/iDefense-IDS-Evasion/PPT_iDefense_IDS_Evasion_20060505.pdf.
706. Eric Peter and Todd Schiller, (2008), A Practical Guide to Honeypots, from https://www.cse.wustl.edu/~jain/cse571-09/ftp/honey/index.html#sec1.4.
707. Krishna Prasad P, (2017), Capturing Attacks on IoT Devices with a multi-purpose IoT Honeypot, from https://security.cse.iitk.ac.in/sites/default/files/15111021.pdf.
708. Nishit Majitha, (2017), Honey-System: Design, Implementation, & Attack Analysis, from https://security.cse.iitk.ac.in/sites/default/files/15111024.pdf.
709. (2015), Bypassing Modern XSS WAF Filters, from http://raijee1337.blogspot.in/2015/09/bypassing-modern-xss-waf-filters.html.
710. (2006), Intrusion Detection System (IDS) Evasion, An iDefense Security Report, from http://www.complianceandprivacy.com/WhitePapers/iDefense-IDS-Evasion/iDefense_IDSEvasion_20060510.pdf.
711. Intrusion detection system evasion techniques, from https://en.wikipedia.org/wiki/Intrusion_detection_system_evasion_techniques.
712. Kyöstilä, Tomi, (2014), The effectiveness of evasion techniques against intrusion prevention systems, from https://aaltodoc.aalto.fi/bitstream/handle/123456789/13589/master_Ky%c3%b6stil%c3%a4_Tomi_2014.pdf?sequence=1&isAllowed=y.
713. Laurent Oudot, Thorsten Holz, (2004), Defeating Honeypots: Network issues, Part 1, from https://www.symantec.com/connect/articles/defeating-honeypots-network-issues-part-1.
714. Laurent Oudot, Thorsten Holz, (2004), Defeating Honeypots: Network Issues, Part 2, from https://www.symantec.com/connect/articles/defeating-honeypots-network-issues-part-2.
715. Owen Watson, (2016), Countering Attack Deception Techniques, from http://slideplayer.com/slide/4495884/.
716. Jon Oberheide and Manish Karir, Honeyd Detection via Packet Fragmentation, from https://www.merit.edu/wp-content/uploads/2016/01/Honeyd_Detection.pdf.
717. Examining Different Types of Intrusion Detection Systems, from https://www.dummies.com/computers/operating-systems/windows-xp-vista/examining-different-types-of-intrusion-detection-systems/.
718. Polymorphic virus, from https://www.trendmicro.com/vinfo/us/security/definition/Polymorphic-virus.
719. (2018), Network Design: Firewall, IDS/IPS, from https://resources.infosecinstitute.com/network-design-firewall-idsips/#gref.
720. What is an Intrusion Prevention System?, from https://www.paloaltonetworks.com/cyberpedia/what-is-an-intrusion-prevention-system-ips.
721. Honeypot (computing), from https://en.wikipedia.org/wiki/Honeypot_(computing).
722. Roger Galobardes, (2018), Learn how Easy is to Bypass Firewalls using DNS Tunneling (and also how to Block it), from https://medium.com/@galolbardes/learn-how-easy-is-to-bypass-firewalls-using-dns-tunneling-and-also-how-to-block-it-3ed652f4a000.
723. (2019), DNS Tunneling Techniques in Cyberattacks, from https://www.andreafortuna.org/2019/01/16/dns-tunneling-techniques-in-cyberattacks/.
724. Ron Lifinski, (2018), How Hackers Use DNS Tunneling to Own Your Network, from https://www.cynet.com/blog/how-do-hackers-use-dns-tunneling-to-own-your-network/.

Module 13: Hacking Web Servers

725. Web Parameter Tampering, from https://owasp.org/www-community/attacks/Web_Parameter_Tampering.
726. Meier, John D, (2010), Configuring and organizing server security information, from http://www.freepatentsonline.com/7712137.html.
727. Internet Security, from https://en.wikipedia.org/wiki/Internet_security.
728. Securing applications, from https://www.slideshare.net/florinc/application-security-1831714.
729. Robert Auger, (2009), Server Misconfiguration, from http://projects.webappsec.org/w/page/13246959/Server%20Misconfiguration.
730. (2009), Cache Poisoning, from https://owasp.org/www-community/attacks/Cache_Poisoning.
731. Improving Web Application Security: Threats and Countermeasures, from https://docs.microsoft.com/en-us/previous-versions/msp-n-p/ff649874(v=pandp.10)?redirectedfrom=MSDN.

732. Rick Rosato, (2014), Best Practices for Applying Service Packs, Hotfixes and Security Patches, from https://docs.microsoft.com/en-us/previous-versions/msp-n-p/ff649874(v=pandp.10)?redirectedfrom=MSDN.
733. J.D. Meier, Alex Mackman, Michael Dunner, Srinath Vasireddy, Ray Escamilla, and Anandha Murukan, (2010), Securing Your Web Server, from https://docs.microsoft.com/en-us/previous-versions/msp-n-p/ff648653(v=pandp.10)?redirectedfrom=MSDN.
734. Web Server Security and Database Server Security, from https://www.acunetix.com/websitesecurity/webserver-security/.
735. Windows IIS Server hardening checklist, from https://searchsecurity.techtarget.com/feature/Windows-IIS-server-hardening-checklist.
736. IIS Web Server Security, from https://www.acunetix.com/websitesecurity/iis-security/.
737. Checklist: Securing Your Web Server, from https://docs.microsoft.com/en-us/previous-versions/msp-n-p/ff648198(v=pandp.10)?redirectedfrom=MSDN.
738. HTTP Response Splitting, Web Cache Poisoning Attacks, and Related Topics, from http://www.ouah.org/whitepaper_httpresponse.pdf.
739. Hacking Web Servers, from https://www.scribd.com/doc/35607686/hacking-Module-11.
740. Terms used by Microsoft to describe the various software updates released by it, from https://www.thewindowsclub.com/terms-used-by-microsoft-to-describe-the-various-software-updates-released-by-it.
741. Directory Traversal Attacks, from https://www.acunetix.com/websitesecurity/directory-traversal/.
742. Jason Chan, (2006), Essentials of Patch Management Policy and Practice, from http://www.patchmanagement.org/pmessentials.asp.
743. Frank Kargl, Jörn Maier, Stefan Schlott, Michael Weber, Protecting Web Servers from Distributed Denial of Service Attacks, from http://www10.org/cdrom/papers/409/.
744. Radu State, (2008), Hacking Web2, from http://www.aims-conference.org/issnsm-2008/01-WebHacking.pdf.
745. Security issues affecting Apache httpd 2.0.40, from http://www.apacheweek.com/features/security-v2.0.40.
746. Apache HTTP Server 2.2 vulnerabilities, from https://httpd.apache.org/security/vulnerabilities_22.html.
747. Saumil Shah, (2003), One-way Web Hacking, from http://www.net-square.com/_assets/One-way_Web_Hacking.pdf.
748. Shani, Oren, (2010), Definition: WEB-SITES DEFACEMENT, from http://www.freepatentsonline.com/y2010/0107247.html.
749. Bodvoc, (2010), An Overview of a Web Server, from https://bodvoc.wordpress.com/2010/07/02/an-overview-of-a-web-server/.
750. (2009), IIS 7.0 Architecture, from https://www.gandhipritesh.com/2009/05/iis-70-architecture.html.
751. Robert Auger, Server Misconfiguration, from http://projects.webappsec.org/w/page/13246959/Server-Misconfiguration.
752. Insecure Configuration Management, from http://www.owasp.org/index.php/Insecure_Configuration_Management.
753. Robert Auger, HTTP Response Splitting, from http://projects.webappsec.org/w/page/13246931/HTTP-Response-Splitting.
754. HTTP Response Splitting, from https://owasp.org/www-community/attacks/HTTP_Response_Splitting.
755. (2005), Introduction to HTTP Response Splitting, from https://securiteam.com/securityreviews/5WP0E2KFGK.
756. Tunneling protocol, from https://en.wikipedia.org/wiki/Tunneling_protocol.
757. How to hack a Web Server, from https://www.guru99.com/how-to-hack-web-server.html.
758. Siddharth Bhattacharya, (2009), Hacking A Web Site and Secure Web Server Techniques Used, from https://www.slideshare.net/siddharthbhattacharya/hacking-a-web-site-and-secure-web-server-techniques-used.
759. (2014), What is the ultimate goal of hacking a webserver?, from https://security.stackexchange.com/questions/48705/what-is-the-ultimate-goal-of-hacking-a-webserver.
760. DNS Hijacking: What is it and How it Works, from https://www.gohacking.com/dns-hijacking/.
761. Niranjan, (2006), DNS Amplification Attack, from http://nirlog.com/2006/03/28/dns-amplification-attack/.
762. (2009), How to detect if your webserver is hacked and get alerted, from https://www.webdigi.co.uk/blog/2009/how-to-detect-if-your-webserver-is-hacked-and-get-alerted.
763. Amit Klein, (2004), HTTP Response Splitting, Web Cache Poisoning Attacks, from http://www.ouah.org/whitepaper_httpresponse.pdf.
764. Kevin Beaver, Top hacker tricks to exploit SQL Server systems, from https://searchsqlserver.techtarget.com/tip/Ten-hacker-tricks-to-exploit-SQL-Server-systems?Offer=SQLwnha217.

765. (2007), Windows IIS server hardening checklist, from https://searchsecurity.techtarget.com/feature/Windows-IIS-server-hardening-checklist.

766. Web Server, from https://www.tutorialspoint.com/internet_technologies/web_servers.htm.

767. Web server, from https://en.wikipedia.org/wiki/Web_server.

768. Rajiv, (2015), Explain Web servers operation and general server characteristics? , from https://eduladder.com/viewquestions/975/Explain-Web-servers-operation-and-general-server-characteristics.

769. Virtual hosting, from https://en.wikipedia.org/wiki/Virtual_hosting.

770. Margaret Rouse, (2006), Virtual Hosting, from https://whatis.techtarget.com/definition/virtual-hosting.

771. Tim Fisher, (2019), What is a Web Proxy?, from https://www.lifewire.com/what-is-web-proxy-3481607.

772. Web proxy servers, from https://en.wikipedia.org/wiki/Proxy_server#Web_proxy_servers.

773. Addison Wesley Longman, 2003, Web Server Operation, from http://web.cs.wpi.edu/~kal/courses/awt/lab6/wwwch11servlets.PDF.

774. Web Servers Hacking, from http://www.certiology.com/computing/certified-ethical-hacker/ethical-hacking-study-guide/how-to-hack-a-web-server.html.

775. (2019), What is the Server Side Request Forgery Vulnerability & How to Prevent It?, from https://www.netsparker.com/blog/web-security/server-side-request-forgery-vulnerability-ssrf/.

776. Ian Muscat, (2019), What is Server Side Request Forgery (SSRF)?, from https://www.acunetix.com/blog/articles/server-side-request-forgery-vulnerability/.

777. Server-side request forgery (SSRF), from https://portswigger.net/web-security/ssrf.

778. Windows Patch Management Best Practices, from https://www.gfi.com/patch-management/windows-patch-management.

779. Web Server Attacks and Countermeasures, from https://sites.google.com/a/pccare.vn/it/security-pages/web-server-attacks-and-countermeasures.

780. System Security Guidelines, from http://www.gswan.gov.in/PDF/Computer-System-Security-Guidelines-4-14122017.pdf.

781. Monitor and Detect Who is Making Changes to Your Directories and Network Shares in Real-time, from https://directorymonitor.com/.

782. (2019), DNS Hijacking: How to Identify and Protect Against it, from https://securitytrails.com/blog/dns-hijacking.

Module 14: Hacking Web Applications

783. (2006), ISYOUR WEBSITE HACKABLE, from http://www.acunetix.com/vulnerability-scanner/wvsbrochure.pdf.

784. (2006), The 21 Primary Classes of Web Application Threats, from www.netcontinuum.com/securityCentral/TopThreatTypes/index.cfm.

785. IMS General Web Services Security Profile, from http://www.imsglobal.org/gws/gwsv1p0/imsgws_securityProfv1p0.html.

786. Path Traversal and URIs, from https://phucjimy.wordpress.com/category/document-security/.

787. Code Injection, from https://owasp.org/www-community/attacks/Code_Injection.

788. J. Howard Beales, III, (2004), OWASP Web Application Security Vulnerabilities Top Ten List, from www.owasp.org/images/c/ce/OWASP_Top_Ten_2004.doc.

789. Connection String Injection Attacks, from https://docs.microsoft.com/en-us/dotnet/framework/data/adonet/connection-string-builders.

790. Connection String Parameter Pollution Attacks, from https://blackhat.com/presentations/bh-dc-10/Alonso_Chema/Blackhat-DC-2010-Alonso-Connection-String-Parameter-Pollution-wp.pdf.

791. Session Prediction, from https://www.owasp.org/index.php?title=Session_Prediction&setlang=en.

792. Robert Auger, (2010), Buffer Overflow, from http://projects.webappsec.org/w/page/13246916/Buffer-Overflow.

793. Managed Web Application Firewall, from https://www.secureworks.com/resources/ds-managed-waf.

794. Do you write secure code?, from https://www.slideshare.net/yuvalgo/do-you-write-secure-code-by-erez-metula.

795. Web Parameter Tampering, from https://owasp.org/www-community/attacks/Web_Parameter_Tampering.

796. (2009), Path Traversal, from https://owasp.org/www-community/attacks/Path_Traversal.

797. LDAP Injection & BLIND LDAP Injection, from https://www.blackhat.com/presentations/bh-europe-08/Alonso-Parada/Whitepaper/bh-eu-08-alonso-parada-WP.pdf.

798. Parameter Manipulation, from https://www.cgisecurity.com/owasp/html/ch11s04.html.

799. (2016), Cross-site Scripting (XSS), from https://owasp.org/www-community/attacks/xss/.

800. Robert "RSnake" Hansen, (2014), XSS Filter Evasion Cheat Sheet, from https://owasp.org/www-community/xss-filter-evasion-cheatsheet.

801. Cross-Site Request Forgery (CSRF) Attack Lab, from http://www.cis.syr.edu/~wedu/seed/Labs_12.04/Web/Web_CSRF_Elgg/Web_CSRF_Elgg.pdf.

802. CHRIS, (2004), Cross-Site Request Forgeries, from http://www.shiflett.org/articles/cross-site-request-forgeries.

803. Web application Attack: DOS and DDOS attack, from http://funwhichuwant.blogspot.in/2012/10/webapplication-attack-dos-and-ddos.html.

804. Preetish Panda, 2009, Web Application Vulnerabilities, from https://www.slideshare.net/technoplex/web-application-vulnerabilities.

805. Yusuf Motiwala, (2007), Attacking XML Security, from https://www.slideshare.net/yusufmotiwala/attacking-xml-security.

806. Managing Web Services, from https://docs.oracle.com/cd/E19316-01/820-4335/gbbjk/index.html.

807. Web Services Hacking And Hardening, from https://www.slideshare.net/rnewton/web-services-hacking-and-hardening.

808. Shreeraj Shah, (2007), Advanced Web Services Hacking, from https://www.slideshare.net/shreeraj/advanced-web-services-hacking.

809. Testing for HTTP Splitting/Smuggling (OWASP-DV-016), from https://www.owasp.org/index.php/Testing_for_HTTP_Exploit.

810. Testing for SQL Wildcard Attacks (OWASP-DS-001), from https://www.owasp.org/index.php/Testing_for_SQL_Wildcard_Attacks_(OWASP-DS-001).

811. Testing for DoS User Specified Object Allocation (OWASP-DS-004), from https://www.owasp.org/index.php/Testing_for_DoS_User_Specified_Object_Allocation_(OWASP-DS-004).

812. Testing for Storing too Much Data in Session (OWASP-DS-008), from https://www.owasp.org/index.php/Testing_for_Storing_too_Much_Data_in_Session_(OWASP-DS-008).

813. Testing for Naughty SOAP Attachments, from https://nilminus.wordpress.com/web-application-penetration-testing/web-services-testing/testing-for-naughty-soap-attachments/.

814. Testing for AJAX (OWASP-AJ-002), from Testing for AJAX (OWASP-AJ-002), from https://www.owasp.org/index.php/Testing_for_AJAX_(OWASP-AJ-002).

815. Common Web-Based Applications Attacks, from http://www.applicure.com/Common_Web_Based_Applications_Attacks#2._Injection_Flaws.

816. Paper -- Cross Site Scripting, from http://www.technicalinfo.net/papers/CSS.html.

817. Robert J. Shimonski, (2002), Hacking techniques, from https://www.ibm.com/developerworks/security/library/s-crack/password_cracking.html

818. Sarah Granger, (2002), A Guide To Better Password Practices, from https://www.symantec.com/connect/articles/simplest-security-guide-better-password-practices.

819. Bad Password Examples, from http://www.spy-hill.com/~myers/help/Passwords.html.

820. Password cracker, from https://searchfinancialsecurity.techtarget.com/definition/password-cracker.

821. John, HTTP Authentication: Basic and Digest Access Authentication, from https://www.ietf.org/rfc/rfc2617.txt.

822. Authentication, Authorization, and Access Control, from https://httpd.apache.org/docs/2.4/howto/auth.html.

823. The Cross-Site Scripting (XSS) FAQ, from https://www.cgisecurity.com/xss-faq.html.

824. Quick Security Reference - Cross-Site Scripting.docx, from http://download.microsoft.com/download/E/E/7/EE7B9CF4-6A59-4832-8EDE-B018175F4610/Quick%20Security%20Reference%20-%20Cross-Site%20Scripting.docx.

825. Web Application Penetration Testing, from https://www.owasp.org/index.php/Web_Application_Penetration_Testing.

826. Jeff Orloff, The Big Website Guide to a Hacking Attack, from http://www.applicure.com/blog/big-website-guide-to-a-hacking-attack.

827. What is Cross-Site Scripting (XSS)?, from http://www.applicure.com/blog/what-is-cross-site-scripting.

828. LDAP Filters, from http://www.selfadsi.org/ldap-filter.htm.

829. Paul Lee, (2002), Cross-site scripting, from https://www.ibm.com/developerworks/tivoli/library/s-csscript/.

830. Cross Site Scripting Prevention Cheat Sheet, from https://owasp.org/www-project-cheat-sheets/cheatsheets/Cross_Site_Scripting_Prevention_Cheat_Sheet.html.

831. Amit Klein, (2005), DOM Based Cross Site Scripting or XSS of the Third Kind, from http://www.webappsec.org/projects/articles/071105.shtml.
832. Philip Tellis, (2010), Common Security Mistakes in Web Applications, from https://www.smashingmagazine.com/2010/10/common-security-mistakes-in-web-applications/.
833. J.D. Meier, Alex Mackman, Michael Dunner, Srinath Vasireddy, Ray Escamilla and Anandha Murukan, (2003), Improving Web Application Security: Threats and Countermeasures, from https://docs.microsoft.com/en-us/previous-versions/msp-n-p/ff649874(v=pandp.10)?redirectedfrom=MSDN.
834. Alex Homer, (2009), Components and Web Application Architecture, from https://docs.microsoft.com/en-us/previous-versions/windows/it-pro/windows-2000-server/bb727121(v=technet.10)?redirectedfrom=MSDN.
835. Unvalidated Input, from https://wiki.owasp.org/index.php/Unvalidated_Input.
836. Kevin Beaver, The importance of input validation, from https://searchsoftwarequality.techtarget.com/tip/The-importance-of-input-validation.
837. (2010), Validating Input, from https://developer.apple.com/library/content/documentation/Security/Conceptual/SecureCodingGuide/Articles/ValidatingInput.html.
838. Code injection, from https://en.wikipedia.org/wiki/Code_injection.
839. Injection Prevention Cheat Sheet, from https://owasp.org/www-project-cheat-sheets/cheatsheets/Injection_Prevention_Cheat_Sheet.html.
840. Remote file inclusion, from https://en.wikipedia.org/wiki/File_inclusion_vulnerability.
841. Robert Auger, (2011), LDAP Injection, from http://projects.webappsec.org/w/page/13246947/LDAP%20Injection.
842. Testing for LDAP Injection (OWASP-DV-006), from https://www.owasp.org/index.php?title=Testing_for_LDAP_Injection_(OTG-INPVAL-007)&redirect=no.
843. Cross-site scripting, from https://en.wikipedia.org/wiki/Cross-site_scripting.
844. DOM Based XSS, from https://owasp.org/www-community/attacks/DOM_Based_XSS.
845. Phil Haack, (2009), CSRF Attacks and Web Forms, from https://haacked.com/archive/2009/04/02/csrf-webforms.aspx/.
846. (2004), Cross-Site Request Forgeries, from http://shiflett.org/articles/cross-site-request-forgeries.
847. Robert Auger, (2010), The Cross-Site Request Forgery (CSRF/XSRF) FAQ, from https://www.cgisecurity.com/csrf-faq.html.
848. Application Denial of Service, from https://www.owasp.org/index.php/Application_Denial_of_Service.
849. Broken Authentication and Session Management, from https://www.owasp.org/index.php/Broken_Authentication_and_Session_Management.
850. (2009), Buffer Overflow, from https://owasp.org/www-community/vulnerabilities/Buffer_Overflow.
851. Robert Auger, (2010), Brute Force, from http://projects.webappsec.org/w/page/13246915/Brute-Force.
852. (2009), Session Prediction, from https://owasp.org/www-community/attacks/Session_Prediction.
853. Robert Auger, (2010), XPath Injection, from http://projects.webappsec.org/w/page/13247005/XPath-Injection.
854. (2009), XPATH Injection, from https://owasp.org/www-community/attacks/XPATH_Injection.
855. Akshay Jindal, Web Application Attack: Injection flaws Attack, from http://funwhichuwant.blogspot.in/search?updated-max=2012-10-12T23:01:00-07:00&max-results=10&reverse-paginate=true&start=79&by-date=false.
856. Preetish Panda, (2009), Web Application Vulnerabilities, from https://www.slideshare.net/technoplex/web-application-vulnerabilities.
857. Gursev Singh Kalra, (2012), Attacking CAPTCHAs for Fun and Profit, from https://www.owasp.org/images/0/03/ASDC12-Attacking_CAPTCHAs_for_Fun_and_Profit.pdf.
858. Gursev Singh Kalra, (2012), Random Security, from http://gursevkalra.blogspot.com/2012/03/captcha-re-riding-attack.html.
859. netbiosX, (2013), Detecting Web Application Firewalls, from https://pentestlab.blog/2013/01/13/detecting-web-application-firewalls/.
860. (2013), How To Hack: Wafw00f Tutorial – Web Application Firewall Detection Tool, from http://ultimatepeter.com/how-to-hack-wafw00f-tutorial-web-application-firewall-detection-tool.
861. Testing for HTTP Splitting/Smuggling (OTG-INPVAL-016), from https://www.owasp.org/index.php/Testing_for_HTTP_Splitting/Smuggling_(OTG-INPVAL-016).

862. Testing for SQL Wildcard Attacks (OWASP-DS-001), from https://wiki.owasp.org/index.php/Testing_for_SQL_Wildcard_Attacks_(OWASP-DS-001).
863. Testing for DoS User Specified Object Allocation (OWASP-DS-004), from https://wiki.owasp.org/index.php/Testing_for_DoS_User_Specified_Object_Allocation_(OWASP-DS-004).
864. Testing for Storing too Much Data in Session (OWASP-DS-008), from https://wiki.owasp.org/index.php/Testing_for_Storing_too_Much_Data_in_Session_(OWASP-DS-008).
865. Testing for AJAX (OWASP-AJ-002), from https://www.owasp.org/index.php?title=Testing_for_AJAX_(OWASP-AJ-002)&setlang=es.
866. GNU General Public License, from https://www.gnu.org/software/wget/manual/wget.html#GNU-Free-Documentation-License.
867. David Larochelle, Statically Detecting Likely Buffer Overflow Vulnerabilities, http://lclint.cs.virginia.edu/usenix01.html.
868. Dawn Song, Web Security, from http://inst.eecs.berkeley.edu/~cs161/fa08/Notes/nov10-xss.pdf.
869. what is a cookie?, how cookie works, from http://www.perlservices.net/en/faq/cookies.shtml#two.
870. CWE-79: Failure to Preserve Web Page Structure ('Cross-site Scripting'), from http://cwe.mitre.org/data/definitions/79.html.
871. Caleb Sima, Security at the Next Level, from http://wp.bitpipe.com/resource/org_1015278444_799/webappwhitepaper2.pdf.
872. Input Validation Attacks, from https://www.insecure.in/input_validation.asp.
873. (2002), Securing Your Web Applications: Anatomy of a Web Attack, from http://www.ebizq.net/topics/security/features/1713.html.
874. Password Cracker, from https://searchfinancialsecurity.techtarget.com/definition/password-cracker?track=malware_glossary.
875. HTTP cookie, from https://en.wikipedia.org/wiki/HTTP_cookie.
876. Abodiford, (2014), Sensitive Data Exposure, from https://www.slideshare.net/abodiford/sensitive-data-exposure.
877. (2017), XXE Injection Attacks – XML External Entity Vulnerability With Examples, from https://www.darknet.org.uk/2017/10/xxe-injection-attacks-xml-external-entity-vulnerability-examples/.
878. Alex Coleman, User Authentication and Access Control in a Web Application, from https://selftaughtcoders.com/user-authentication-access-control-web-application/.
879. Web Application Attack Trends, from https://www.ptsecurity.com/upload/corporate/ww-en/analytics/Web-Application-Attack-Trends-2017-eng.pdf.
880. What is Web Services?, from https://www.javatpoint.com/restful-web-services-what-is-web-services.
881. Server-side JavaScript code injection, from https://portswigger.net/kb/issues/00100d00_server-side-javascript-code-injection.
882. Server Side JS Injection, from https://ckarande.gitbooks.io/owasp-nodegoat-tutorial/content/tutorial/a1_-_server_side_js_injection.html.
883. Joel Scambray, Vincent Liu, and Caleb Sima, (2011), Hacking Exposed Web Applications, 3rd edition, New York, McGraw-Hill.
884. Log Injection, from https://owasp.org/www-community/attacks/Log_Injection.
885. (2019), HTML Injection Tutorial: Types & Prevention with Examples, from https://www.softwaretestinghelp.com/html-injection-tutorial/.
886. HTML Injection, from https://www.imperva.com/learn/application-security/html-injection/.
887. Peter Yaworski, (2017), Web Hacking 101, Lean Pub.
888. (2019), CRLF Injection and HTTP Response Splitting Vulnerability, from https://www.netsparker.com/blog/web-security/crlf-http-header/.
889. What are CRLF Injection Attacks, from https://www.acunetix.com/websitesecurity/crlf-injection/.
890. Frame Injection, from https://secapps.com/vulndb/frame-injection.
891. Ziyahan Albeniz, (2019), Frame Injection Attacks, from https://www.netsparker.com/blog/web-security/frame-injection-attacks/.
892. Frame Injection, from http://doc.cenzic.com/sadoc9x14ba847/CPL0001506.htm.

893. Lokesh Gupta, Java AES 256 Encryption Decryption Example, from https://howtodoinjava.com/security/aes-256-encryption-decryption/.

894. Broken Authentication and Session Management, from https://hdivsecurity.com/owasp-broken-authentication-and-session-management.

895. JavaScript Hijacking, from https://vulncat.fortify.com/en/detail?id=desc.dataflow.java.javascript_hijacking.

896. Amrita Mitra, (2017), What is JSON Hijacking or JavaScript Hijacking?, from https://www.thesecuritybuddy.com/vulnerabilities/what-is-json-hijacking-or-javascript-hijacking/.

897. Mathy Vanhoef and Frank Piessens, (2015), RC4 NOMORE, from https://www.rc4nomore.com/.

898. (2016), Timing Attacks in the Modern Web, from https://tom.vg/2016/08/browser-based-timing-attacks/.

899. Catalin Cimpanu, (2019), New browser attack lets hackers run bad code even after users leave a web page, from https://www.zdnet.com/article/new-browser-attack-lets-hackers-run-bad-code-even-after-users-leave-a-web-page/.

900. Chris Love, (2008), Breaking Down the MarioNet Service Worker Attack? Can Your Website Be Compromised?, from https://love2dev.com/pwa/marionet-attack/.

901. Panagiotis Papadopoulos, Panagiotis Ilia, Michalis Polychronakis, Evangelos P. Markatos, Sotiris Ioannidis, and Giorgos Vasiliadis, Master of Web Puppets: Abusing Web Browsers for Persistent and Stealthy Computation, from https://www.ndss-symposium.org/wp-content/uploads/2019/02/ndss2019_01B-2_Papadopoulos_paper.pdf.

902. Zbigniew Banach, (2019), Clickjacking Attacks: What They are and How to Prevent Them, from https://www.netsparker.com/blog/web-security/clickjacking-attacks/.

903. What is clickjacking, from https://www.imperva.com/learn/application-security/clickjacking/.

904. Chandan Kumar, (2019), Default Port Numbers You Need to Know as a Sysadmin, from https://geekflare.com/default-port-numbers/.

905. (2020), List of TCP and UDP Port Numbers, from https://en.wikipedia.org/wiki/List_of_TCP_and_UDP_port_numbers.

906. (2019), Scan Websites for Interesting Directories & Files with Gobuster, from https://null-byte.wonderhowto.com/how-to/scan-websites-for-interesting-directories-files-with-gobuster-0197226/.

907. Enumerating directories used by popular web applications and servers, from https://nmap.org/nsedoc/scripts/http-enum.html.

908. (2017), How to enumerate webserver directories using Nmap in Kali Linux, from https://ourcodeworld.com/articles/read/416/how-to-enumerate-webserver-directories-using-nmap-in-kali-linux .

909. Zbigniew Banach, (2019), XSS Filter Evasion, from https://www.netsparker.com/blog/web-security/xss-filter-evasion/.

910. Do Son, Everythings do to bypass XSS filter, from https://securityonline.info/bypass-xss-filter/.

911. Dafydd Stuttard and Marcus Pinto, (2011), The Web Application Hacker's Handbook, 2nd edition, Indianapolis, Wiley Publishing.

912. (2019), Password Reset Vulnerability (Poisoning), from https://www.acunetix.com/blog/articles/password-reset-poisoning/.

913. Prosper Otemuyiwa, (2016), What is SAML?, from https://auth0.com/blog/how-saml-authentication-works/.

914. Single Sign-On, from https://auth0.com/docs/sso/current.

915. Jem Jensen, (2017), Attacking SSO: Common SAML Vulnerabilities and Ways to Find Them, from https://blog.netspi.com/attacking-sso-common-saml-vulnerabilities-ways-find/ .

916. Macro Mosaic, (2018), Hack SAML Single Sign-on with Burp Suite, from https://null-byte.wonderhowto.com/how-to/hack-saml-single-sign-with-burp-suite-0184405/.

917. Local File Inclusion (LFI), from https://xapax.gitbooks.io/security/content/local_file_inclusion.html .

918. (2017), Web Application Penetration Testing, from https://www.exploit-db.com/docs/english/40992-web-app-penetration-testing---local-file-inclusion-(lfi).pdf.

919. SQL Server 2019 connection strings, from https://www.connectionstrings.com/sql-server-2019/.

920. (2015), SOAPAction Spoofing, from https://www.ws-attacks.org/index.php/SOAPAction_Spoofing.

921. Prakash Dhatti, (2017), OWASP Top 10: Penetration Testing with SOAP Service and Mitigation, from https://blog.securelayer7.net/owasp-top-10-penetration-testing-soap-application-mitigation/.

922. (2015), WS-Addressing spoofing, from https://www.ws-attacks.org/index.php/WS-Addressing_spoofing.

923. Gutiérrez and Carlos A, WS-Address Spoofing, from https://books.google.as/books?id=dU4E_e8mtDcC&pg=PA222&lpg=PA222&dq=WS-

Addressing+spoofing&source=bl&ots=A-YYZ5tLjQ&sig=ACfU3U3v6YrEwkwmeS-rMr0vgA4pSE_rOQ&hl=en&sa=X&ved=2ahUKEwiP4rzj5ObmAhWx4zgGHcieAm84ChDoATAAegQIBxAB#v=onepage&q=WS-Addressing%20spoofing&f=false.

924. Web API, from https://www.grapecity.com/componentone/web-api.
925. (2019), Web API, from https://en.wikipedia.org/wiki/Web_API.
926. Introduction to web APIs, from https://developer.mozilla.org/en-US/docs/Learn/JavaScript/Client-side_web_APIs/Introduction.
927. Nyma Malik, (2017), The Difference Between REST and SOAP APIs, from https://dzone.com/articles/difference-between-rest-and-soap-api.
928. Saquib Rizwan, (2017), An Introduction to RESTful APIs, from https://dzone.com/articles/an-introduction-to-restful-apis.
929. REST API and its Component, from https://www.360logica.com/blog/rest-api-and-its-component/.
930. Margaret Rouse, RESTful API (REST API), from https://searchapparchitecture.techtarget.com/definition/RESTful-API.
931. Types of APIs, from https://rapidapi.com/blog/types-of-apis/.
932. Matthew Guay, (2018), What Are Webhooks?, from https://zapier.com/blog/what-are-webhooks/.
933. (2018), What are Webhooks? Easy Explanation & Tutorial, from https://snipcart.com/blog/what-are-webhooks-explained-example.
934. (2017), Webhook vs API: What's the Difference Between them?, from http://techtales.co/2017/08/07/webhook-vs-api-whats-difference/.
935. (2019), What is API Security?, from https://owasp.org/www-project-api-security/.
936. Chris Romeo, (2019), OWASP API Security Top 10: Get your dev team up to speed, from https://techbeacon.com/security/owasp-api-security-top-10-get-your-dev-team-speed.
937. Jason Skowronski, (2019), Common API Vulnerabilities and How to Secure Them, from https://blog.papertrailapp.com/common-api-vulnerabilities-and-how-to-secure-them/.
938. Thomas Bush, (2019), 5 Common API Vulnerabilities (and How to Fix Them), from https://nordicapis.com/5-common-api-vulnerabilities-and-how-to-fix-them/.
939. Anand Srinivasan, (2017), Top 5 Vulnerabilities in APIs, from https://datafloq.com/read/top-5-vulnerabilities-in-apis/2876.
940. MrBrijesh, (2015), Web API – Security Review – How to Hack an API and Get Away with it, from https://mrbrijesh.wordpress.com/2015/04/13/web-api-security-review-how-to-hack-an-api-and-get-away-with-it/.
941. Ole Lensmar, (2014), API Security Testing – How to Hack an API and Get Away with It, from https://smartbear.com/blog/test-and-monitor/api-security-testing-how-to-hack-an-api-part-1/.
942. Daniel Tomescu, XML Based Attacks, from https://www.owasp.org/images/5/58/XML_Based_Attacks_-_OWASP.pdf.
943. Bernard Harguindeguy, (2019), How Does Your API Security Stand Up Against the 3 Most Common Attacks?, from https://www.programmableweb.com/news/how-does-your-api-security-stand-against-3-most-common-attacks/sponsored-content/2019/01/03.
944. What is Credential Stuffing?, from https://www.cloudflare.com/learning/bots/what-is-credential-stuffing/.
945. (2020), Credential stuffing, from https://en.wikipedia.org/wiki/Credential_stuffing.
946. Bernard Harguindeguy, (2019), How Does Your API Security Stand Up Against the 3 Most Common Attacks?, from https://www.programmableweb.com/news/how-does-your-api-security-stand-against-3-most-common-attacks/sponsored-content/2019/01/03.
947. Mitchell Anicas, (2014), An Introduction to OAuth 2, from https://www.digitalocean.com/community/tutorials/an-introduction-to-oauth-2.
948. Dhaval Kapil, (2017), Attacking the OAuth Protocol , from https://dhavalkapil.com/blogs/Attacking-the-OAuth-Protocol/.
949. OAuth 2.0 Authorization Framework, from https://auth0.com/docs/protocols/oauth2.
950. Gaurang Bhatnagar, (2018), OAuth Attacks, from https://www.slideshare.net/OWASPdelhi/pentesting-rest-apis-by-gaurang-bhatnagar.
951. Kristopher Sandoval, (2018), 5 Ways to Hack an API (and How to Defend), from https://nordicapis.com/5-ways-to-hack-an-api-and-how-to-defend/.
952. Gaurang Bhatnagar, (2019), Bypassing IDOR via Parameter Pollution, from https://medium.com/@0xgaurang/case-study-bypassing-idor-via-parameter-pollution-78f7b3f9f59d.

953. Agathoklis Prodromou, (2016), An Introduction to Web-shells – Part 1, from https://www.acunetix.com/blog/articles/introduction-web-shells-part-1/.
954. (2020), Web shell, from https://en.wikipedia.org/wiki/Web_shell.
955. Remote Access: WebShells, from https://www.rsa.com/content/dam/en/solution-brief/asoc-threat-solution-series-webshells.pdf.
956. (2018), How to Upload a PHP Web Shell Using Weevely to Get Backdoor Access, from https://www.shellvoide.com/hacks/how-to-upload-php-web-shell-using-weevely-to-get-backdoor-access/.
957. (2019), Web Shell, from https://attack.mitre.org/techniques/T1100/.
958. Agathoklis Prodromou, (2016), Detection and Prevention – An Introduction to Web-Shells – Final Part, from https://www.acunetix.com/blog/articles/detection-prevention-introduction-web-shells-part-5/.
959. (2014), Protecting Your APIs Against Attack and Hijack: Secure your enterprise applications for mobile, the cloud and open Web, from https://docs.broadcom.com/docs/protecting-your-apis-against-attack-and-hijack.
960. API Security Architecture, from https://www.axway.com/en/products/api-management/manage-apis/security-architecture.
961. Radware, (2019), How to Prevent Real-Time API Abuse, from https://blog.radware.com/security/applicationsecurity/2019/04/how-to-prevent-real-time-api-abuse/.
962. (2018), Basic Steps for API Security, from https://www.immuniweb.com/blog/basic-steps-for-api-security.html.
963. State of API Security, from https://www.soapui.org/learn/security/state-of-api-security.html.
964. Securing webhooks, from https://anymail.readthedocs.io/en/stable/tips/securing_webhooks/.
965. Webhook Signature Hash Validation and Security, from https://ordapad.help/docs/webhook-signature-hash-validation.
966. Yakir Perlin, How to validate Cloudinary webhooks Signature?, from https://support.cloudinary.com/hc/en-us/articles/115001302471-How-to-validate-Cloudinary-webhooks-signature-.
967. Best practices for using webhooks, from https://stripe.com/docs/webhooks/best-practices.
968. Webhook Best Practices, from https://apidocs.leadsquared.com/webhook-best-practices/.
969. Best Practices, from https://developer.bigcommerce.com/api-docs/getting-started/best-practices.
970. Web Application Security Testing, from https://www.techopedia.com/definition/29826/web-application-security-testing.
971. Sherif Koussa, (2018), SAST, DAST, IAST, and RASP, from https://www.softwaresecured.com/what-do-sast-dast-iast-and-rasp-mean-to-developers/.
972. Thomas Scanlon, (2018), 10 Types of Application Security Testing Tools: When and How to Use Them, from https://insights.sei.cmu.edu/sei_blog/2018/07/10-types-of-application-security-testing-tools-when-and-how-to-use-them.html.
973. Aditya Kakrania, (2017), Manual or Automated Application Security Testing: What's More Effective?, from https://blog.securityinnovation.com/manual-or-automated-application-security-testing-whats-more-effective.
974. Sharon Solomon, (2015), Application Security Testing – Automated Vs Manual, from https://www.checkmarx.com/2015/05/19/application-security-testing-automated-vs-manual/.
975. Adam Sedgewick, Murugiah Souppaya, and Karen Scarfone, (2015), Guide to Application Whitelisting, from https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-167.pdf.
976. Calyptix, (2017), Application Whitelisting: What it is and why it's good, from https://www.calyptix.com/top-threats/application-whitelisting-good/.
977. Margaret Rouse, (2019), Application whitelisting, from https://searchsecurity.techtarget.com/definition/application-whitelisting.
978. Allow or Block Access to Websites, from https://support.google.com/chrome/a/answer/7532419?hl=en.
979. Paul Rubens, (2018), How to Prevent SQL Injection Attacks, from https://www.esecurityplanet.com/threats/how-to-prevent-sql-injection-attacks.html.
980. Protecting Against SQL Injection, from https://www.hacksplaining.com/prevention/sql-injection.
981. What is the SQL Injection Vulnerability & How to Prevent it?, from https://www.netsparker.com/blog/web-security/sql-injection-vulnerability/.
982. Primary Defenses, from https://cheatsheetseries.owasp.org/cheatsheets/OS_Command_Injection_Defense_Cheat_Sheet.html.

983. Serge Truth, (2011), How to Test for Command Injection, from https://blog.securityinnovation.com/blog/2011/06/how-to-test-for-command-injection.html.

984. David McMillen, (2016), The Importance of Thwarting Command Injection Attacks, from https://www.ibm.com/downloads/cas/DRNKZBR1.

985. LDAP and LDAP Injection/Prevention, from https://www.geeksforgeeks.org/ldap-ldap-injectionprevention/.

986. (2018), Understanding and Defending Against LDAP Injection Attacks, from https://ldap.com/2018/05/04/understanding-and-defending-against-ldap-injection-attacks/.

987. (2017), Top 10 Common Web Attacks: The First Steps to Protect Your Website, from https://www.vpnmentor.com/blog/top-10-common-web-attacks/.

988. Lucero Davalos Vizcarra, (2019), Top 10 Web Security Vulnerabilities to Watch Out for in 2019, from https://cai.tools.sap/blog/top-10-web-security-vulnerabilities-to-watch-out-for-in-2019/.

989. Cross-Site Scripting (XSS), from https://phpsecurity.readthedocs.io/en/latest/Cross-Site-Scripting-(XSS).html.

990. A Positive XSS Prevention Model, from https://cheatsheetseries.owasp.org/cheatsheets/Cross_Site_Scripting_Prevention_Cheat_Sheet.html.

991. Directory Traversal, from https://portswigger.net/web-security/file-path-traversal.

992. Preventing directory traversal , from https://www.hacksplaining.com/prevention/directory-traversal.

993. (2013), Unvalidated Redirects and Forwards, from https://hdivsecurity.com/owasp-unvalidated-redirects-and-forwards.

994. (2017), What Is an Unvalidated Redirect/Forward? , from https://www.sitelock.com/blog/how-to-mitigate-unvalidated-redirects-forwards/.

995. (2018), Safety Tips for Watering Hole Attacks, from https://www.techadvisory.org/2018/04/safety-tips-for-watering-hole-attacks/.

996. Laurencem, (2018), 5 Ways to Defend Against Watering Hole Attacks, from https://www.itsasap.com/2018/07/31/5-ways-defend-watering-hole-attacks/.

997. Resources that Need to be Protected from CSRF Vulnerability, from https://cheatsheetseries.owasp.org/cheatsheets/Cross-Site_Request_Forgery_Prevention_Cheat_Sheet.html.

998. CSRF Attacks: Anatomy, Prevention, and XSRF Tokens, from https://www.acunetix.com/websitesecurity/csrf-attacks/.

999. (2006), Cookie Poisoning Prevention in ASP.NET, from https://searchsoftwarequality.techtarget.com/answer/Cookie-poisoning-prevention-in-ASPNET.

1000. Nathan Rossiter, (2014), Common Web Application Attacks and How to Prevent Them, from https://www.business2community.com/crisis-management/common-web-application-attacks-prevent-0949592 .

1001. Alex Mitchell, (2019), How Does One Defend Against Session Hijacking?, from https://hakin9.org/how-does-one-defend-against-session-hijacking/.

1002. Username Enumeration, from https://guides.codepath.com/websecurity/Username-Enumeration.

1003. Username enumeration, from https://docs.kentico.com/spaces/flyingpdf/pdfpageexport.action?pageId=68881105.

1004. Patrick Laverty, (2017), What Is User Enumeration?, from https://blog.rapid7.com/2017/06/15/about-user-enumeration/.

1005. (2015), Username Enumeration Techniques and their Value, from https://www.nccgroup.trust/uk/about-us/newsroom-and-events/blogs/2015/june/username-enumeration-techniques-and-their-value/.

1006. Abhinav Mishra, (2015), Abusing Password Reset Functionality to Steal User Data (Part – I), from https://www.tothenew.com/blog/abusing-password-reset-part-1/.

1007. John P. Mello Jr, What is Runtime Application Self-Protection (RASP)?, from https://techbeacon.com/security/what-runtime-application-self-protection-rasp.

1008. Bria Grangard, (2019), RASP 101: What Is Runtime Application Self-Protection?, from https://blog.rapid7.com/2019/09/04/rasp-101-what-is-runtime-application-self-protection/.

1009. (2016), Defending Web Apps: WAFs versus RASPs, from https://warroom.rsmus.com/wafs-vs-rasps/.

1010. Govindraj Basatwar, (2019), What Is Runtime Application Self-Protection (RASP)?, from https://www.appsealing.com/what-is-runtime-application-self-protection/.

1011. (2019), Bug bounty program, from https://en.wikipedia.org/wiki/Bug_bounty_program.

Module 15: SQL Injection

1012. (2008), Preventing SQL Injections in Online Applications: Study, Recommendations and Java Solution Prototype Based on the SQL DOM, from http://mirror.kioss.undip.ac.id/pustaka-bebas/library-sw-hw/linux-1/security/WebGoat/OWASP/OWASP-AppSecEU08-Janot.pdf.

1013. San-Tsai Sun, (2007), Classification of SQL Injection Attacks, from http://courses.ece.ubc.ca/412/term_project/reports/2007-fall/Classification_of_SQL_Injection_Attacks.pdf.

1014. Victor Chapela, Advanced SQL Injection, from https://www.slideshare.net/amiable_indian/advanced-sql-injection.

1015. (2012), Advanced SQL Injection, from http://blogs.pages.kr/1341.

1016. Dmitry Evteev, (2009), Advanced SQL Injection, from https://www.slideshare.net/devteev/advanced-sql-injection-eng.

1017. (2005), SQL injection, from http://searchsqlserver.techtarget.com/feature/SQL-injection.

1018. What is SQL Injection?, from https://www.secpoint.com/sql-injection.html.

1019. Rise in SQL Injection Attacks Exploiting Unverified User Data Input, from https://docs.microsoft.com/en-us/security-updates/SecurityAdvisories/2008/954462.

1020. (2006), Injection Protection, from https://docs.microsoft.com/en-us/previous-versions/sql/legacy/aa224806(v=sql.80)?redirectedfrom=MSDN.

1021. Understanding SQL Injection, from https://tools.cisco.com/security/center/resources/sql_injection.

1022. SQL Injection, from https://owasp.org/www-community/attacks/SQL_Injection.

1023. San-Tsai Sun, Ting Han Wei, and Stephen Liu, Sheung Lau, (2007), Classification of SQL Injection Attacks, from http://courses.ece.ubc.ca/412/term_project/reports/2007-fall/Classification_of_SQL_Injection_Attacks.pdf.

1024. Krzysztof Kotowicz, (2010), SQL Injection: Complete walkthrough (not only) for PHP developers, from https://www.slideshare.net/kkotowicz/sql-injection-complete-walktrough-not-only-for-php-developers.

1025. (2007), INTRODUCTION, from http://isea.nitk.ac.in/publications/web.pdf.

1026. Dmitry Evteev, (2009), Advanced SQL Injection, from http://www.ptsecurity.com/download/PT-devteev-Advanced-SQL-Injection-ENG.zip.

1027. Cameron Hotchkies, (2004), Blind SQL Injection Automation Techniques, from http://www.blackhat.com/presentations/bh-usa-04/bh-us-04-hotchkies/bh-us-04-hotchkies.pdf.

1028. SQL Injection, from https://docs.microsoft.com/en-us/previous-versions/sql/sql-server-2008-r2/ms161953(v=sql.105)?redirectedfrom=MSDN.

1029. SQL INJECTION, from http://www.authorstream.com/Presentation/useful-155975-sql-injection-hacking-computers-22237-education-ppt-powerpoint/.

1030. Ferruh Mavituna, (2007), SQL Injection Cheat Sheet, from https://www.netsparker.com/blog/web-security/sql-injection-cheat-sheet/.

1031. K. K. Mookhey and Nilesh Burghate, (2004), Detection of SQL Injection and Cross-site Scripting Attacks, from https://www.symantec.com/connect/articles/detection-sql-injection-and-cross-site-scripting-attacks.

1032. Debasish Das, Utpal Sharma, and D.K. Bhattacharyya, (2010), An Approach to Detection of SQL Injection Attack Based on Dynamic Query Matching, from https://www.ijcaonline.org/journal/number25/pxc387766.pdf.

1033. (2010), Quick Security Reference: SQL Injection, from http://download.microsoft.com/download/E/E/7/EE7B9CF4-6A59-4832-8EDE-B018175F4610/Quick%20Security%20Reference%20-%20SQL%20Injection.docx.

1034. Ferruh Mavituna, One Click Ownage, Adventures of a lazy pentester, from https://www.owasp.org/images/8/8e/One_Click_Ownage-Ferruh_Mavituna.pdf.

1035. Alexander Kornbrust, (2009), ODTUG - SQL Injection Crash Course for Oracle Developers, from http://www.red-database-security.com/wp/OOW2009_sql_crashcourse_for_developers.pdf.

1036. William G.J. Halfond, Jeremy Viegas, and Alessandro Orso, (2006), A Classification of SQL Injection Attack Techniques and Countermeasures, from https://www.cc.gatech.edu/~orso/papers/halfond.viegas.orso.ISSSE06.presentation.pdf.

1037. Injection Flaws - OWASP, from https://www.owasp.org/index.php/Injection_Flaws.

1038. Victor Chapela, (2005), Advanced SQL Injection, from http://www.owasp.org/images/7/74/Advanced_SQL_Injection.ppt.

1039. (2010), SQL Injection, from https://docs.microsoft.com/en-us/previous-versions/sql/sql-server-2008/ms161953(v=sql.100)?redirectedfrom=MSDN.

1040. Blind SQL Injection, from http://www.evilsql.com/main/page1.php.

1041. SQL Injection, from https://www.w3schools.com/sql/sql_injection.asp.

1042. SQL Injection Cheat Sheet & Tutorial: Vulnerabilities & How to Prevent SQL Injection Attacks, from https://www.veracode.com/security/sql-injection.

1043. Types of SQL Injection (SQLi), from https://www.acunetix.com/websitesecurity/sql-injection2/.

1044. Everything You Need to Know About SQL Injection Attacks & Types, SQLi Code Example, Variations, Vulnerabilities & More, from http://www.firewall.cx/general-topics-reviews/web-application-vulnerability-scanners/1207-how-sql-injection-attacks-work-examples.html.

1045. Hack2Secure, (2017), Understanding SQL Injection Attacks, from https://www.hack2secure.com/blogs/understanding-sql-injection-attacks.

1046. Using Comments to Simplify SQL Injection, from https://www.sqlinjection.net/comments/.

1047. SQL Injection Cheat Sheet, from https://www.netsparker.com/blog/web-security/sql-injection-cheat-sheet/#InlineComments.

1048. (2017), SQL Injection Tutorial , from https://www.w3resource.com/sql/sql-injection/sql-injection.php.

1049. Nuno Loureiro, (2010), Advanced SQL Injection: Attacks, from https://www.slideshare.net/nuno.loureiro/advanced-sql-injection-attacks.

1050. Types of SQL Injection Attacks, from http://hwang.cisdept.cpp.edu/swanew/Text/SQL-Injection.htm.

1051. Time-Based Blind SQL Injection using Heavy Query, from https://www.sqlinjection.net/heavy-query/.

1052. (2017), SQL Injection Bypassing WAF, from https://www.owasp.org/index.php/SQL_Injection_Bypassing_WAF.

1053. (2013), WAF SQL evasion using HPF technique, from https://security.stackexchange.com/questions/34498/waf-sql-evasion-using-hpf-technique.

1054. (2009), HTTP Parameter Fragmentation (HPF) is one of the methods to bypass security filters in web applications, from http://blog.ptsecurity.com/2009/12/http-parameter-fragmentation-hpf-is-one.html.

1055. Luca Carettoni and Stefano di Paola, HTTP Parameter Pollution, from https://www.owasp.org/images/b/ba/AppsecEU09_CarettoniDiPaola_v0.8.pdf.

1056. Steve Friedl, (2017), SQL Injection Attacks by Example, from http://www.unixwiz.net/techtips/sql-injection.html.

1057. Simone Quatrini and Marco Rondini, "Blind Sql Injection with Regular Expressions Attack", from https://www.exploit-db.com/docs/english/17397-blind-sql-injection-with-regular-expressions-attack.pdf.

1058. (2010), Hacking website using SQL Injection -step by step guide, from https://breakthesecurity.cysecurity.org/2010/12/hacking-website-using-sql-injection-step-by-step-guide.html.

1059. Chandrakant Nial, How to Find Admin Login Panel of a Website, from http://www.darksite.co.in/2013/04/how-to-find-admin-login-panel-of.html.

1060. SQL Injection techniques, from https://www.oratechinfo.co.uk/sql_injection.html.

1061. PL/SQL Attacks, from https://www.sqlinjection.net/advanced/pl-sql/.

1062. PL/SQL Security Vulnerabilities and Language Overview, from https://www.checkmarx.com/plsql-security-vulnerabilities-platform-overview/.

1063. MacGyver, (2013), Create server backdoors using SQL Injection, from http://macgyverdev.blogspot.in/2013/02/create-server-backdoors-using-sql.html.

1064. (2012), Creating Backdoors Using SQL Injection, from https://resources.infosecinstitute.com/backdoor-sql-injection/#gref.

1065. (2011), Leveraging a shell from SQL injection, from https://security.stackexchange.com/questions/6919/leveraging-a-shell-from-sql-injection.

1066. SQL Injection Attack, from https://shodhganga.inflibnet.ac.in/bitstream/10603/123504/7/chapter%202.pdf.

1067. Advanced evasion techniques for defeating SQL injection Input validation mechanisms, from https://jeannotclement.wordpress.com/2011/01/16/advanced-evasion-technique/.

1068. Evasion Techniques, from http://books.gigatux.nl/mirror/apachesecurity/0596007248/apachesc-CHP-10-SECT-8.html.

1069. Intrusion detection system evasion techniques, from https://en.wikipedia.org/wiki/Intrusion_detection_system_evasion_techniques.

1070. Do Son, (2019), SQL injection: 9 Ways to Bypass Web Application Firewall, from https://securityonline.info/sql-injection-9-ways-bypass-web-application-firewall/.

1071. SQL Injection Bypass CheatSheet, from http://www.0xby.com/873.html.

1072. System Tables and Views, from https://docs.oracle.com/database/timesten-18.1/TTSYS/systemtables.htm#TTSYS722.

1073. Grant Tables, from https://dev.mysql.com/doc/refman/8.0/en/grant-tables.html

1074. (2017), Remove references to undocumented system tables, from https://docs.microsoft.com/en-us/sql/sql-server/install/remove-references-to-undocumented-system-tables?view=sql-server-2014.

1075. (2017), sys.syslogins (Transact-SQL), from https://docs.microsoft.com/en-gb/sql/relational-databases/system-compatibility-views/sys-syslogins-transact-sql?view=sql-server-2017.

1076. Victor Chapela, (2005), Advanced SQL Injection, from https://www.owasp.org/images/7/74/Advanced_SQL_Injection.ppt.

1077. sys.sysdatabases (Transact-SQL), from https://docs.microsoft.com/en-us/sql/relational-databases/system-compatibility-views/sys-sysdatabases-transact-sql?view=sql-server-2017.

1078. sys.sysobjects (Transact-SQL), from https://docs.microsoft.com/en-us/sql/relational-databases/system-compatibility-views/sys-sysobjects-transact-sql?view=sql-server-2017.

1079. sys.syscolumns (Transact-SQL), from https://docs.microsoft.com/en-us/sql/relational-databases/system-compatibility-views/sys-syscolumns-transact-sql?view=sql-server-2017.

1080. sys.sysservers (Transact-SQL), from https://docs.microsoft.com/en-us/sql/relational-databases/system-compatibility-views/sys-sysservers-transact-sql?view=sql-server-2017.

1081. xp_cmdshell Server Configuration Option, from https://docs.microsoft.com/en-us/sql/database-engine/configure-windows/xp-cmdshell-server-configuration-option?view=sql-server-2017.

1082. (2012), SQL Injection through HTTP Headers, from https://resources.infosecinstitute.com/sql-injection-http-headers/#gref.

1083. Nikos Danopoulos, (2019), X Forwarded for SQL injection, from https://outpost24.com/blog/X-forwarded-for-SQL-injection.

1084. Header Field Definitions, from https://www.w3.org/Protocols/rfc2616/rfc2616-sec14.html.

1085. (2019), Intrusion Prevention, from https://fortiguard.com/encyclopedia/ips/41512.

1086. Exfiltrating Data from MS SQL Server Via DNS, from http://pentestmonkey.net/blog/mssql-dns.

1087. Mike Saunders, (2018), SQLi Data Exfiltration via DNS, from https://www.redsiege.com/blog/2018/11/sqli-data-exfiltration-via-dns/.

1088. (2014), DNS Exfiltration with SQL Injection, from https://blog.safebuff.com/2016/06/14/DNS-exfiltration-with-SQL-Injection/.

1089. SQL Injection: Vulnerabilities & How to Prevent SQL Injection Attacks, from https://www.veracode.com/security/sql-injection.

1090. Defeating SQL Injection IDS Evasion, from https://www.giac.org/paper/gcia/1231/defeating-sql-injection-ids-evasion/109964.

1091. (2018), How to Avoid Detection & Bypass Defenses, from https://null-byte.wonderhowto.com/how-to/sql-injection-101-avoid-detection-bypass-defenses-0184918/.

1092. Service overview and network port requirements for Windows, from https://support.microsoft.com/en-in/help/832017/service-overview-and-network-port-requirements-for-windows.

1093. Thomas Sermpinis, Web Application Hacking Advanced SQL Injection and Data Store Attacks, Vol 12, from Book.

Module 16: Hacking Wireless Networks

1094. Jari Arkko, Vesa Torvinen, Aki Niemi, (2002), HTTP Authentication with EAP, from http://www.arkko.com/publications/draft-torvinen-http-eap-01.txt.

1095. Peter Loshin, (2019), Defending against the most common wireless network attacks, from http://searchsecurity.techtarget.com/feature/A-list-of-wireless-network-attacks.

1096. Ajay Kumar Gupta, (2010), Comment: Rogue Access Point Setups on Corporate Networks, from https://www.infosecurity-magazine.com/opinions/comment-rogue-access-point-setups-on-corporate/.

1097. Bluetooth Security Risks and Tips to Prevent Security Threats, from https://www.brighthub.com/computing/smb-security/articles/30045.aspx.

1098. Cisco Unified Wireless Network Architecture—Base Security Features, from https://www.cisco.com/c/en/us/td/docs/solutions/Enterprise/Mobility/secwlandg20/sw2dg/ch4_2_SPMb.pdf.

1099. Chris Weber and Gary Bahadu, (2009), Wireless Networking Security, from https://docs.microsoft.com/en-us/previous-versions/windows/it-pro/windows-xp/bb457019(v=technet.10)?redirectedfrom=MSDN.

1100. (2006), How to Cheat at Securing a Wireless Network, from https://www.sciencedirect.com/science/article/pii/B9781597490870500572.

1101. Lisa Phifer, (2005), Eliminating interference thru Wi-Fi spectrum analysis, from https://searchmobilecomputing.techtarget.com/tip/Eliminating-interference-thru-Wi-Fi-spectrum-analysis.
1102. Understanding WiFi Hotspots... from https://www.scambusters.org/wifi.html.
1103. (2009), How 802.11 Wireless Works, from https://docs.microsoft.com/en-us/previous-versions/windows/it-pro/windows-server-2003/cc757419(v=ws.10)?redirectedfrom=MSDN.
1104. TKIP (Temporal Key Integrity Protocol), from https://www.tech-faq.com/tkip-temporal-key-integrity-protocol.html.
1105. Kevin Beaver and Peter T. Davis, Understanding WEP Weaknesses, from https://www.dummies.com/programming/networking/understanding-wep-weaknesses/.
1106. Eric Geier, (2010), 7 Things Hackers Hope You Don't Know, from https://www.esecurityplanet.com/views/article.php/3891716/7-Things-Hackers-Hope-You-Dont-Know.htm.
1107. Rogue Wireless Access Point, from https://www.tech-faq.com/rogue-wireless-access-point.html.
1108. How to War Drive, from https://www.wikihow.com/War-Drive.
1109. ALFRED LOO, (2009), Security Threats of Smart Phones and Bluetooth, from http://www.aaronfrench.com/coursefiles/ucommerce/Loo_2009.pdf.
1110. Prabhaker Mateti, Hacking Techniques in Wireless Networks, from https://web1.cs.wright.edu/~pmateti/InternetSecurity/Lectures/WirelessHacks/Mateti-WirelessHacks.htm.
1111. Bradley Mitchell, (2020) Wired vs. Wireless Networking, from https://www.lifewire.com/wired-vs-wireless-networking-816352.
1112. Bradley Mitchell, (2019), Wireless Standards - 802.11b 802.11a 802.11g and 802.11n, from https://www.lifewire.com/wireless-standards-802-11a-802-11b-g-n-and-802-11ac-816553.
1113. Wi-Fi Protected Access, from https://searchmobilecomputing.techtarget.com/definition/Wi-Fi-Protected-Access.
1114. WPA (Wi-Fi Protected Access), from https://www.tech-faq.com/wpa-wi-fi-protected-access.shtml.
1115. Paul Arana, (2006), Benefits and Vulnerabilities of Wi-Fi Protected Access 2 (WPA2), from https://cs.gmu.edu/~yhwang1/INFS612/Sample_Projects/Fall_06_GPN_6_Final_Report.pdf.
1116. Wireless Hacking, from https://www.darknet.org.uk/category/wireless-hacking/.
1117. Gary Wollenhaupt, How Cell Phone Jammers work, from https://electronics.howstuffworks.com/cell-phone-jammer1.htm.
1118. Brian R. Miller and Booz Allen Hamilton, (2002), Issues in Wireless security, from https://www.acsac.org/2002/case/wed-c-330-Miller.pdf.
1119. Jonathan Hassell, (2004), Wireless Attacks and Penetration Testing, from https://www.symantec.com/connect/articles/wireless-attacks-and-penetration-testing-part-1-3.
1120. Martin Beck and TU-Dresden, (2008), Practical attacks against WEP and WPA, from http://dl.aircrack-ng.org/breakingwepandwpa.pdf.
1121. Chris Hurley, Finding cloaked access points, (Chapter 9), from https://books.google.co.in/books?id=wGJhDNspE3wC&pg=PA333&lpg=PA333&dq=cloaked+access+point&source=bl&ots=ZDkHSykDNV&sig=1sLKIx-1ZcqkhUdr1WpFaqYczyI&hl=en&ei=V8R2Ss35Oo2e6gP59viqCw&sa=X&oi=book_result&ct=result&redir_esc=y#v=onepage&q=cloaked%20access%20point&f=false.
1122. (2002), Wireless Scanning Wardriving / Warchalking, from https://www.it-observer.com/wireless-scanning-wardriving-warchalking.html.
1123. Wireless Network, from http://www.hackingtheuniverse.com/information-security/attack-vs-defense/attack-vs-defense-on-an-organizational-scale/5-wireless-network.
1124. Michael Roche, (2007), Wireless Attack Tools, from https://www.cse.wustl.edu/~jain/cse571-07/ftp/wireless_hacking.pdf.
1125. Protecting your wireless network from hacking, from http://www.businessknowledgesource.com/technology/protecting_your_wireless_network_from_hacking_025027.html.
1126. Agustina, J. V. Peng Zhang, and Kantola, (2003), Performance evaluation of GSM handover traffic in a GPRS/GSM network, from https://ieeexplore.ieee.org/document/1214113?isnumber=27298&arnumber=1214113&count=217&index=21.
1127. Service set identifier, from https://searchmobilecomputing.techtarget.com/definition/service-set-identifier.
1128. Antenna Cabling Guide, from http://wireless.gumph.org/content/3/12/011-antenna-cabling.html.
1129. The Wireless Intrusion detection system, from http://www.forum-intrusion.com/widz_design.pdf.

1130. Humphrey Cheung, (2005), How To Crack WEP - Part 1: Setup & Network Recon, from https://www.tomsguide.com/us/how-to-crack-wep,review-451.html.

1131. Humphrey Cheung, (2005), How To Crack WEP - Part 2: Performing the Crack, from https://www.tomsguide.com/us/how-to-crack-wep,review-459.html.

1132. Humphrey Cheung, (2005), How To Crack WEP - Part 3: Securing your WLAN, from https://www.tomsguide.com/us/how-to-crack-wep,review-471.html.

1133. Chris Weber and Gary Bahadur, (2009), Wireless Networking Security, from https://docs.microsoft.com/en-us/previous-versions/windows/it-pro/windows-xp/bb457019(v=technet.10)?redirectedfrom=MSDN.

1134. (2009), How 802.11 Wireless Works, from https://docs.microsoft.com/en-us/previous-versions/windows/it-pro/windows-server-2003/cc757419(v=ws.10)?redirectedfrom=MSDN.

1135. Brandon Teska, (2008), How To Crack WPA / WPA2, from https://www.smallnetbuilder.com/wireless/wireless-howto/30278-how-to-crack-wpa--wpa2.

1136. A Comprehensive Review of 802.11 Wireless LAN Security and the Cisco Wireless Security Suite, from https://www.cisco.com/c/en/us/products/collateral/wireless/aironet-1200-access-point/prod_white_paper09186a00800b469f.html.

1137. (2006), How To Crack WEP and WPA Wireless Networks, from http://121space.com/index.php?showtopic=3376.(2006), How To Crack WEP and WPA Wireless Networks, from http://121space.com/index.php?showtopic=3376.

1138. (2009), How to prevent wireless DoS attacks, from https://searchsecurity.techtarget.com/feature/How-to-prevent-wireless-DoS-attacks.

1139. Jim Geier, (2003), Denial of Service a Big WLAN Issue, from https://www.esecurityplanet.com/trends/article.php/2200071/Denial-of-Service-a-Big-WLAN-Issue.htm.

1140. Peter Loshin, (2009), A list of wireless network attacks, from https://searchsecurity.techtarget.com/feature/A-list-of-wireless-network-attacks.

1141. Lisa Phifer, (2009), A wireless network vulnerability assessment checklist, from https://searchsecurity.techtarget.com/feature/A-wireless-network-vulnerability-assessment-checklist.

1142. Lisa Phifer, (2009), Hunting for rogue wireless devices, from https://searchsecurity.techtarget.com/feature/Hunting-for-rogue-wireless-devices.

1143. PreciousJohnDoe, List of Wireless Network Attacks, from http://www.brighthub.com/computing/smb-security/articles/53949.aspx.

1144. Security Disciplines for Objective 3: Detection and Recovery, from https://it.ojp.gov/documents/wirelesssecurity.pdf.

1145. Jim Geier, How to: Sniff Wireless Packets with Wireshark, from http://www.wireless-nets.com/resources/tutorials/sniff_packets_wireshark.html.

1146. Laurent Oudot, (2004), Wireless Honeypot Countermeasures, from https://www.symantec.com/connect/articles/wireless-honeypot-countermeasures.

1147. (2009), Fragmentation Attack, from http://www.aircrack-ng.org/doku.php?id=fragmentation.

1148. Andrei A. Mikhailovsky, Konstantin V. Gavrilenko, and Andrew Vladimirov, (2004), The Frame of Deception: Wireless Man-in-the-Middle Attacks and Rogue Access Points Deployment, from http://www.informit.com/articles/article.aspx?p=353735&seqNum=7.

1149. Renee Oricchio, How to Surf Safely on Public Wi-Fi, from https://www.inc.com/telecom/articles/200707/wifi.html.

1150. What is WiFi, from https://www.scambusters.org/wifi.html.

1151. Trishna Panse and Prashant Panse, (2013), A Survey on Security Threats and Vulnerability attacks on Bluetooth Communication, from http://www.ijcsit.com/docs/Volume%204/Vol4Issue5/ijcsit2013040521.pdf.

1152. How to Bluejack, from https://www.wikihow.com/Bluejack.

1153. John Padgette and Karen Scarfone, (2012), Guide to Bluetooth Security (Draft), from https://csrc.nist.gov/csrc/media/publications/sp/800-121/rev-1/final/documents/draft-sp800-121_rev1.pdf.

1154. Nateq Be-Nazir Ibn Minar and Mohammed Tarique, (2012), BLUETOOTH SECURITY THREATS AND SOLUTIONS: A SURVEY, from http://airccse.org/journal/ijdps/papers/0112ijdps10.pdf.

1155. Lisa Phifer, Wireless network troubleshooting: Connectivity, from https://searchnetworking.techtarget.com/tip/Wireless-network-troubleshooting-Connectivity.

1156. Keijo M.J. Haataja, (2005), Detailed descriptions of new proof-of-concept Bluetooth security analysis tools and new security attacks, from http://www.cs.uku.fi/tutkimus/publications/reports/B-2005-1.pdf.

1157. (2017), What You Should Know About the 'KRACK' WiFi Security Weakness, from https://krebsonsecurity.com/2017/10/what-you-should-know-about-the-krack-wifi-security-weakness/.

1158. Lily Hay Newman, (2017), The 'Secure' Wi-Fi Standard has a Huge, Dangerous Flaw, from https://www.wired.com/story/krack-wi-fi-wpa2-vulnerability/.

1159. Steve Tilson, (2017), WPA2 Key Reinstallation Attack (KRACK) Vulnerability Detection Dashboard, from https://www.tenable.com/sc-dashboards/wpa2-key-reinstallation-attack-krack-vulnerability-detection-dashboard.

1160. Thomas Brewster, (2017), Update Every Device -- This KRACK Hack Kills Your Wi-Fi Privacy, from https://www.forbes.com/sites/thomasbrewster/2017/10/16/krack-attack-breaks-wifi-encryption/#3d9b890e2ba9.

1161. Paul Ducklin, (2017), Wi-Fi at risk from KRACK attacks – here's what to do, from https://nakedsecurity.sophos.com/2017/10/16/wi-fi-at-risk-from-krack-attacks-heres-what-to-do/.

1162. Charlie Osborne and Zack Whittaker, (2017), Here's every patch for KRACK Wi-Fi vulnerability available right now, from http://www.zdnet.com/article/here-is-every-patch-for-krack-wi-fi-attack-available-right-now/.

1163. Michael Heller, (2017), KRACK WPA2 flaw might be more hype than risk, from http://searchsecurity.techtarget.com/news/450428414/KRACK-WPA2-vulnerability-might-be-more-hype-than-risk

1164. (2017), KRACK WPA Vulnerability - Key Reinstallation AttaCK, from https://aircrack-ng.blogspot.in/2017/10/krack-wpa-vulnerability-key.html.

1165. Attacks on EAP Protocols, from http://etutorials.org/Networking/Wireless+lan+security/Chapter+6.+Wireless+Vulnerabilities/Attacks+on+EAP+Protocols/.

1166. Mateusz Buczkowski, (2018), Introduction to Wi-Fi Security, from https://www.grandmetric.com/2018/07/06/ended-wpa3-wi-fi-security-evolution/.

1167. Wireless Security Protocols: WEP, WPA, WPA2 and WPA3, from https://www.cyberpunk.rs/wireless-security-protocols-wep-wpa-wpa2-and-wpa3.

1168. Penny Hoelscher, (2018), What is WPA3, is it secure and should I use it?, from https://www.comparitech.com/blog/information-security/what-is-wpa3/.

1169. Discover Wi-Fi Security, from https://www.wi-fi.org/discover-wi-fi/security.

1170. (2018), WPA3 Explained, from https://medium.com/@reliancegcs/wpa3-explained-wi-fi-is-getting-major-security-update-2b6dca8f3aff.

1171. (2020), Wi-Fi Protected Access, from https://en.wikipedia.org/wiki/Wi-Fi_Protected_Access.

1172. Mark Wyciślik-Wilson, aLTEr: Hackers can spy on your 4G browsing sessions thanks to LTE flaws, from https://betanews.com/2018/06/30/alter-lte-vulnerability/.

1173. (2018), aLTEr: POC Exploit of LTE Layer Two, from https://blog.zimperium.com/alter-poc-exploit-lte-layer-two/.

1174. Nicole Lorenz, (2018), aLTEr: Attack every smartphone via LTE, from https://blog.avira.com/alter-attack-every-smartphone-via-lte/.

1175. Aaron Woland, (2018), Protecting iOS against the aLTEr attacks, from https://www.networkworld.com/article/3287149/protecting-ios-against-the-alter-attacks.html .

1176. (2013), Wireless Attacks Unleashed, from https://resources.infosecinstitute.com/wireless-attacks-unleashed/#gref.

1177. Elahe Fazeldehkordi and Oluwatobi Ayodeji Akanbi, (2016), Wormhole Attack, from https://www.sciencedirect.com/topics/computer-science/wormhole-attack.

1178. Julián Ramírez Gómez, Héctor Fernando Vargas Montoya, and Álvaro León Henao1, (2019), Implementing a Wormhole Attack on Wireless Sensor Networks with XBee S2C Devices, from https://dialnet.unirioja.es/descarga/articulo/6939721.pdf.

1179. (2015), Finding WPS enabled Wifi Networks with Kali Linux Wash, from https://www.hackingtutorials.org/wifi-hacking-tutorials/wps-wifi-networks-with-kali-linux-wash/#prettyPhoto.

1180. Brian Sak and Jilumudi Raghu Ram, (2016), Mastering Kali Linux Wireless Pentesting, Birmingham, Packt Publishing Ltd, from https://github.com/volym3ad/BOOKS/blob/master/Mastering%20Kali%20Linux%20Wireless%20Pentesting.pdf.

1181. (2008), ARP Spoofing, from https://openmaniak.com/ettercap_arp.php.

1182. Zaid Sabih, Creating Fake Access Points with the MANA Toolkit, from https://www.oreilly.com/library/view/learn-ethical-hacking/9781788622059/7b4ce7c7-1a2d-4dc6-8754-799ee9e4d7ba.xhtml.

1183. Arnav Tripathy, (2019), How to Make a Fake Access Point with Mana-Toolkit, from https://medium.com/@arnavtripathy98/how-to-make-a-fake-access-point-with-mana-toolkit-2464c1843d1e.

1184. Balaji N, (2018), Wi-Jacking – New Wifi Attack Allow Accessing Millions of Neighbour's WiFi Without Cracking, from https://gbhackers.com/wi-jacking-wifi-attack/.

1185. Catalin Cimpanu, (2018), Google Fixes Chrome Issue that Allowed Theft of WiFi Logins, from https://www.zdnet.com/article/google-fixes-chrome-issue-that-allowed-theft-of-wifi-logins/.

1186. Gurubaran S, (2019), Pentesting & Crack WPA/WPA2 WiFi Passwords with Wifiphisher by Jamming the WiFi, from https://gbhackers.com/crack-wpawpa2-kali-linux-tutorial/.

1187. Kody, (2015), Get Anyone's Wi-Fi Password Without Cracking Using Wifiphisher, from https://null-byte.wonderhowto.com/how-to/hack-wi-fi-get-anyones-wi-fi-password-without-cracking-using-wifiphisher-0165154/.

1188. Tomáš Foltýn, (2019), WPA3 Flaws May Let Attackers Steal Wi-Fi Passwords, from https://www.welivesecurity.com/2019/04/11/wpa3-flaws-steal-wifi-passwords/.

1189. Catalin Cimpanu, (2019), Dragonblood vulnerabilities disclosed in WiFi WPA3 standard, from https://www.zdnet.com/article/dragonblood-vulnerabilities-disclosed-in-wifi-wpa3-standard/ .

1190. Dan Goodun, (2019), Serious flaws leave WPA3 vulnerable to hacks that steal Wi-Fi passwords, from https://arstechnica.com/information-technology/2019/04/serious-flaws-leave-wpa3-vulnerable-to-hacks-that-steal-wi-fi-passwords/.

1191. Michael Peters, (2019), Dragonblood Vulnerabilities Discovered in WPA3 WiFi Standard, from https://securityboulevard.com/2019/04/dragonblood-vulnerabilities-discovered-in-wpa3-wifi-standard/.

1192. Sergiu Gatlan, (2019), WPA3 Wi-Fi Standard Affected by New Dragonblood Vulnerabilities, from https://www.bleepingcomputer.com/news/security/wpa3-wi-fi-standard-affected-by-new-dragonblood-vulnerabilities/.

1193. Pierluigi Paganini, (2019), WPA3 Attacks Allow Hackers to Hack Wi-Fi Password, from https://securityaffairs.co/wordpress/83653/hacking/wpa3-security-flaws.html.

1194. Daniele Antonioli, Nils Ole Tippenhauer, and Kasper B. Rasmussen, The KNOB is Broken: Exploiting Low Entropy in the Encryption Key Negotiation of Bluetooth BR/EDR, from https://www.usenix.org/conference/usenixsecurity19/presentation/antonioli.

1195. Doug Lynch, (2019), KNOB Attack exploits Bluetooth spec flaw to spy on device connections, from https://www.xda-developers.com/knob-attack-bluetooth-flaw/.

1196. Michael Heller, (2019), KNOB attack puts all Bluetooth devices at risk, from https://searchsecurity.techtarget.com/news/252468914/KNOB-attack-puts-all-Bluetooth-devices-at-risk.

1197. Daniele Antonioli, (2019), About the KNOB Attack, from https://knobattack.com.

1198. Mathy Vanhoef and Eyal Ronen, Dragonblood: Analyzing the Dragonfly Handshake of WPA3 and EAP-pwd, from https://papers.mathyvanhoef.com/dragonblood.pdf.

1199. Trapti Pandey and Pratha Khare, Bluetooth Hacking and its Prevention, from https://www.ltts.com/sites/default/files/resources/pdf/whitepapers/2017-12/Bluetooth-Hacking-and-its-Prevention.pdf.

1200. Art Miller, (2019), How to Protect Yourself from Bluetooth Hacking, from https://www.vectorsecurity.com/blog/how-to-protect-yourself-from-bluetooth-hacking.

Module 17: Hacking Mobile Platforms

1201. Android framework for exploitation, from http://www.xysec.com/afe_manual.pdf.

1202. Sarah Perez, (2010), How to Hack Your Android Phone (and Why You Should Bother), from http://readwrite.com/2010/01/27/how_to_hack_your_android_phone.

1203. (2016), OWASP Mobile Top 10, from https://www.owasp.org/index.php/Projects/OWASP_Mobile_Security_Project_-_Top_Ten_Mobile_Risks.

1204. Security Threat Report 2014, from https://www.sophos.com/en-us/medialibrary/PDFs/other/sophos-security-threat-report-2014.pdf.

1205. wiseGEEK, What is Mobile Phone Spam?, from https://www.wisegeek.com/what-is-mobile-phone-spam.htm.

1206. Murugiah Souppaya and Karen Scarfone, (2013), Guidelines for Managing the Security of Mobile Devices in the Enterprise, from https://csrc.nist.gov/csrc/media/publications/sp/800-124/rev-1/final/documents/draft_sp800-124-rev1.pdf.

1207. Michael Cooney, (2012), 10 common mobile security problems to attack, from https://www.pcworld.com/article/2010278/10-common-mobile-security-problems-to-attack.html.

1208. Shruti Dhapola, (2014), Android is most hacked mobile OS: Here's how to protect your phone, from https://www.firstpost.com/tech/news-analysis/android-malware-increasing-tips-protect-phone-3647981.html.

1209. Network Spoofer - Hacking networks from Android, from http://www.hackedexistence.com/project-networkspoofer.html.

1210. iOS jailbreaking, from https://en.wikipedia.org/wiki/IOS_jailbreaking#Types_of_jailbreaks.
1211. Jeff Benjamin, (2011), Untethered Jailbreak vs. Tethered Jailbreak vs. SemiTethered Jailbreak — What's the Difference?, from https://www.idownloadblog.com/2011/10/22/untethered-jailbreak-vs-tethered-jailbreak-vs-semitethered-jailbreak/.
1212. Jailbreak Your iPhone, iPad And iPod Touch, from https://www.iphoneinformer.com/jailbreak/.
1213. Lisa Phifer, (2013), BYOD security strategies: Balancing BYOD risks and rewards, from https://searchsecurity.techtarget.com/feature/BYOD-security-strategies-Balancing-BYOD-risks-and-rewards.
1214. Stephan Chenette, (2013), Building Custom Android Malware for Penetration Testing, from https://www.slideshare.net/schenette/building-custom-android-malware-brucon-2013.
1215. Al Berg, Best practices for protecting handhelds from mobile malware, from https://searchsecurity.techtarget.com/tip/Best-practices-for-protecting-handhelds-from-mobile-malware.
1216. Sam Bakken, (2017), Defense in Depth: A Layered Approach to Mobile Security with MDM, MAM & Mobile App Vetting, from https://www.nowsecure.com/blog/2017/12/12/defense-in-depth-a-layered-approach-to-mobile-security-with-mdm-mam-mobile-app-vetting/.
1217. (2017), Anatomy of an Android, from https://www.sophos.com/en-us/medialibrary/PDFs/other/sophos-anatomy-of-an-android-infographic.pdf.
1218. Victor Chebyshev, Fedor Sinitsyn, Denis Parinov, Boris Larin, Oleg Kupreev, and Evgeny Lopatin, (2019), IT threat evolution Q2 2019. Statistics, from https://securelist.com/it-threat-evolution-q2-2019-statistics/92053/.
1219. "Agent Smith": The New Virus to Hit Mobile Devices, from https://blog.checkpoint.com/2019/07/10/agent-smith-android-malware-mobile-phone-hack-virus-google/.
1220. (2019), Agent Smith virus hides in WhatsApp, infests 1.5 crore Android phones in India: What is it, should you worry, from https://www.indiatoday.in/technology/news/story/agent-smith-virus-whatsapp-infects-android-phones-in-india-what-is-it-1566668-2019-07-11.
1221. Aviran Hazum, Feixiang He, Inbal Marom, Bogdan Melnykov, and Andrey Polkovnichenko, (2019), Agent Smith: A New Species of Mobile Malware, from https://research.checkpoint.com/agent-smith-a-new-species-of-mobile-malware/.
1222. Pierluigi Paganini, (2016), Researchers hack WhatsApp accounts through SS7 protocol, from https://securityaffairs.co/wordpress/47179/hacking/hacking-ss7-protocol.html.
1223. Samuel Gibbs, (2016), SS7 hack explained: what can you do about it?, from https://www.theguardian.com/technology/2016/apr/19/ss7-hack-explained-mobile-phone-vulnerability-snooping-texts-calls.
1224. Secure your network from SS7 attacks, from https://www.sinch.com/insights/operator-opportunities/ss7/?cn-reloaded=1.
1225. Simjacker, from https://simjacker.com.
1226. Shouvik Das, (2019), Your Data, Location Might be Tracked with This SIM Card Flaw, Without Your Knowledge, from https://www.news18.com/news/tech/your-data-location-might-be-tracked-with-this-sim-card-flaw-without-your-knowledge-2306879.html.
1227. Mohit Kumar, (2019), New SIM Card Flaw Lets Hackers Hijack Any Phone Just by Sending SMS, from https://thehackernews.com/2019/09/simjacker-mobile-hacking.html.
1228. (2019), Device Administration Overview, from https://developer.android.com/guide/topics/admin/device-admin.html.
1229. Ashraf Iftekhar, (2018), How To Test Android Application Security Using Drozer?, from https://medium.com/@ashrafrizvi3006/how-to-test-android-application-security-using-drozer-edc002c5dcac.
1230. Sumit Bhattacharya, Android Penetration Tools Walkthrough Series: Drozer, from https://resources.infosecinstitute.com/android-penetration-tools-walkthrough-series-drozer/#gref.
1231. Romansh Yadav, (2019), Drozer! The Game Changer Tool for Android Pen Testing, from https://blog.securelayer7.net/drozer-the-game-changer-tool-for-android-pentesting/.
1232. Man-in-the-Disk: A New Attack Surface for Android Apps, from https://blog.checkpoint.com/2018/08/12/man-in-the-disk-a-new-attack-surface-for-android-apps/.
1233. Margaret Rouse, (2019), Man-in-the-Disk (MITD) Attack, from https://searchsecurity.techtarget.com/definition/man-in-the-disk-MITD-attack.
1234. Mohammed Tahir, (2019), Latest attack on Android known as "Spearphone Attack", from https://cyberops.in/blog/spearphone-attack/.
1235. Swati Khandelwal, (2019), New Attack Lets Android Apps Capture Loudspeaker Data Without Any Permission, from https://thehackernews.com/2019/07/android-side-channel-attacks.html .

1236. Abeerah Hashim, (2019), Spearphone Attack Allows Android Apps to Listen to Your Loudspeaker Conversations, from https://latesthackingnews.com/2019/07/20/spearphone-attack-allows-android-apps-to-listen-to-your-loudspeaker-conversations/.
1237. Connor Jones, (2019), Android phones vulnerable to advanced SMS phishing attacks, from https://www.itpro.co.uk/security/34334/android-phones-vulnerable-to-advanced-sms-phishing-attacks.
1238. Ravie Lakshmanan, (2019), Hackers are now attacking Android users with advanced SMS phishing techniques, from https://thenextweb.com/security/2019/09/04/hackers-are-now-attacking-android-users-with-advanced-sms-phishing-techniques/.
1239. Aman Bhardwaj, (2019), SSL Pinning: Introduction & Bypass for Android, from https://niiconsulting.com/checkmate/2019/04/ssl-pinning-introduction-bypass-for-android/.
1240. Balaji N, (2019), Bypassing and Disabling SSL Pinning on Android to Perform Man-in-the-Middle Attack, from https://gbhackers.com/bypass-ssl-pinning/.
1241. Gurubaran S, (2019), Newly Discovered Tap 'n Ghost Attack Let Hackers to Remotely Control Android Smartphones, from https://gbhackers.com/tap-n-ghost-attack-remotely-smartphones/.
1242. (2019), Newly discovered Tap 'n Ghost attack can be used to target Android devices, from https://cyware.com/news/newly-discovered-tap-n-ghost-attack-can-be-used-to-target-android-devices-fe81c323.
1243. Ivan Pisarev, (2019), Gustuff: Weapon of Mass Infection, from https://www.group-ib.com/blog/gustuff.
1244. Sven Taylor, (2018), How to Secure Your Android Device in 5 Simple Steps, fromhttps://restoreprivacy.com/secure-android-device/.
1245. Michael Simon, (2019), How to Secure, Protect, and Completely Lock Down Your Android Phone, from https://www.pcworld.com/article/3332211/secure-android-phone.html.
1246. Steven J. and Vaughan-Nichols, (2018), The 10 best ways to secure your Android phone, from https://www.zdnet.com/article/the-ten-best-ways-to-secure-your-android-phone/.
1247. (2020), iOS jailbreaking, from https://en.wikipedia.org/wiki/IOS_jailbreaking#Types_of_jailbreaks.
1248. Roy Iarchy, (2018), iOS Trustjacking – A Dangerous New iOS Vulnerability, from https://www.symantec.com/blogs/feature-stories/ios-trustjacking-dangerous-new-ios-vulnerability.
1249. Joshua Long, (2018), iOS Trustjacking: How Attackers can Hijack Your iPhone, from https://www.intego.com/mac-security-blog/ios-trustjacking-how-attackers-can-hijack-your-iphone/.
1250. (2018), iOS Trustjacking Protection with EMM, from https://arsenb.wordpress.com/2018/04/25/ios-trustjacking-protection-with-emm/.
1251. Lewis Painter, (2019), iPhone Security Tips: How to Protect Your Phone from Hackers, from https://www.macworld.co.uk/how-to/iphone/iphone-security-tips-3638233/.
1252. (2019), 5 Easy Ways to Protect Your iPhone and Privacy in 2020 FREE, from https://www.vpnmentor.com/blog/protect-privacy-iphone/.
1253. Ken Hess, (2014), 10 BYOD policy guidelines for a secure work environment, from https://techtalk.gfi.com/10-byod-policy-guidelines-for-a-secure-work-environment/.
1254. OWASP Mobile Top 10, from https://owasp.org/www-project-mobile-top-10/#tab=Top_10_Mobile_Controls.
1255. Rohit Bhardwaj, Enterprise Ethical Mobile Hacking Using Owasp Top 10 2017, from https://developermarch.com/mds/downloadPDF/MODS17_Enterprise%20Ethical%20Mobile%20Hacking%20Using%20OWASP%20Top%2010%202017_Rohit.pdf.
1256. Tampering and Reverse Engineering, from https://mobile-security.gitbook.io/mobile-security-testing-guide/general-mobile-app-testing-guide/0x04c-tampering-and-reverse-engineering.
1257. Srinivas, Introduction to Reverse Engineering, from https://resources.infosecinstitute.com/android-hacking-and-security-part-18-introduction-to-reverse-engineering/#gref.

Module 18: IoT and OT Hacking

1258. Margaret Rouse, (2016), Internet of Things (IoT), from https://internetofthingsagenda.techtarget.com/definition/Internet-of-Things-IoT.
1259. Bernadette Johnson, How the Internet of Things Works, from https://computer.howstuffworks.com/internet-of-things.htm.
1260. (2016), The Pros and Cons of IoT, from http://www.humavox.com/blog/pros-cons-iot/.
1261. (2015), How IoT Works – An Overview of the Technology Architecture, from https://www.embitel.com/blog/embedded-blog/how-iot-works-an-overview-of-the-technology-architecture-2.

1262. Internet of Things: Explained, from http://www.carritech.com/news/internet-of-things/.

1263. Dr. Gaurav Bajpai, Middleware for Internet of Things, from http://wireless.ictp.it/rwanda_2015/presentations/Middleware_IoT.pdf.

1264. M2M/IoT Sector Map, from http://www.beechamresearch.com/article.aspx?id=4.

1265. Vasanth Ganesan, (2016), Video meets the Internet of Things, from https://www.mckinsey.com/industries/technology-media-and-telecommunications/our-insights/video-meets-the-internet-of-things.

1266. (2015), 11 Internet of Things (IoT) Protocols You Need to Know About, from https://www.rs-online.com/designspark/eleven-internet-of-things-iot-protocols-you-need-to-know-about.

1267. Stan Schneider, (2013), Understanding The Protocols Behind The Internet Of Things, from https://www.electronicdesign.com/technologies/iot/article/21798493/understanding-the-protocols-behind-the-internet-of-things.

1268. Internet of things, from https://en.wikipedia.org/wiki/Internet_of_things#Trends_and_characteristics.

1269. Anupama Kaushik, (2016), IOT-An Overview, from https://www.ijarcce.com/upload/2016/march-16/IJARCCE%20264.pdf.

1270. Karen Rose, Scott Eldridge, Lyman Chapin, (2015), The Internet of Things: An Overview, from https://www.internetsociety.org/wp-content/uploads/2017/08/ISOC-IoT-Overview-20151221-en.pdf.

1271. Bruce Byfield, (2016), The Internet of Things: 7 Challenges, from https://www.datamation.com/data-center/the-internet-of-things-7-challenges.html.

1272. Aritra Sarkhel, (2016), 5 challenges to Internet of Things, from https://tech.economictimes.indiatimes.com/news/internet/5-challenges-to-internet-of-things/52700940.

1273. Robbie Mitchell, (2015), 5 challenges of the Internet of Things, from https://blog.apnic.net/2015/10/20/5-challenges-of-the-internet-of-things/.

1274. Azita Esmaili, (2015), The opportunities and threats of the Internet of Things, from https://www.techworld.com/startups/opportunities-threats-of-internet-of-things-3598417/.

1275. Charlie Ashton, (2015), Is IoT a Threat or an Opportunity for Service Providers?, from https://www.sdxcentral.com/articles/contributed/iot-threat-opportunity-service-providers-charlie-ashton/2015/06/.

1276. Avantika Monnappa, (2018), TOGAF and the Internet of Things, from https://www.simplilearn.com/togaf-applications-in-internet-of-things-iot-article.

1277. Tessel Renzenbrink, (2014), Internet of Things Poses an Unprecedented Privacy Risk, from https://www.elektormagazine.com/articles/internet-of-things-poses-an-unprecedented-privacy-risk.

1278. (2016), Top IoT Vulnerabilities, from https://www.owasp.org/index.php/Top_IoT_Vulnerabilities.

1279. (2015), IoT Attack Surface Areas, from https://www.owasp.org/index.php/IoT_Attack_Surface_Areas.

1280. Jaibeer Malik, (2017), IOT Security: Attack surface areas, Vulnerabilities & Considerations, from https://jaibeermalik.wordpress.com/2017/01/05/iot-security-attack-surface-areas-vulnerabilities-considerations/.

1281. Security for IoT, from https://www.infineon.com/cms/en/applications/smart-card-and-security/iot-security/?redirId=59655#!overview.

1282. Masato Terada, Naoko, and Naoko Ohnishi, (2017), HIRT-PUB16003: Cyber-attacks Using IoT Devices, http://www.hitachi.com/hirt/publications/hirt-pub16003/index.html.

1283. APNIC, (2017), IoT - the Next Wave of DDoS Threat Landscape, from https://www.slideshare.net/apnic/iot-the-next-wave-of-ddos-threat-landscape?qid=b1d633e5-2d40-4151-b3ec-91d93be094ea&v=&b=&from_search=6.

1284. Jaikumar Vijayan, (2014), Target attack shows danger of remotely accessible HVAC systems, from https://www.computerworld.com/article/2487452/cybercrime-hacking/target-attack-shows-danger-of-remotely-accessible-hvac-systems.html.

1285. Paul Roberts, (2012), FBI Issued Alert over July Attack on HVAC System, from https://securityledger.com/2012/12/fbi-issued-alert-over-july-attack-on-hvac-system/.

1286. Erez Metula, (2016), Hacking The IoT (Internet of Things) - PenTesting RF Operated Devices, from https://www.owasp.org/images/2/29/AppSecIL2016_HackingTheIoT-PenTestingRFDevices_ErezMetula.pdf.

1287. Jerry Hildenbrand, (2017), Let's talk about Blueborne, the latest Bluetooth vulnerability, from https://www.androidcentral.com/lets-talk-about-blueborne-latest-bluetooth-vulnerability.

1288. (2017), The Attack Vector “BlueBorne” Exposes Almost Every Connected Device, from https://www.armis.com/blueborne/.

1289. Blueborne Attack Threatens IoT Devices, from https://www.pindrop.com/blog/blueborne-attack-threatens-iot-devices/.

1290. Kim Zetter, (2016), Inside the Cunning, Unprecedented Hack of Ukraine's Power Grid, from https://www.wired.com/2016/03/inside-cunning-unprecedented-hack-ukraines-power-grid/.

1291. Taylor Edginton, Find the IP Address of any IoT Device, from https://www.pingplotter.com/wisdom/article/finding-my-smart-outlet.html.

1292. (2017) Hacking Zigbee Devices, from https://blog.attify.com/hack-iot-devices-zigbee-sniffing-exploitation/.

1293. (2016), IoT Devices Easily Hacked to be Backdoors: Experiment, from https://www.securityweek.com/iot-devices-easily-hacked-be-backdoors-experiment.

1294. Scott Craig, (2016), Telnet: An Attacker's Gateway to the IoT, from https://securityintelligence.com/telnet-an-attackers-gateway-to-the-iot.

1295. (2016), Emulating and Exploiting Firmware binaries – Offensive IoT Exploitation series, from https://resources.infosecinstitute.com/emulating-and-exploiting-firmware-binaries-offensive-iot-exploitation-series/.

1296. OWASP Top 10 Internet of Things Vulnerability Categories, from https://www.owasp.org/images/8/8e/Infographic-v1.jpg.

1297. (2016), IoT Framework Assessment, from https://www.owasp.org/index.php/IoT_Framework_Assessment.

1298. (2019), MQTT v5.0 now an official OASIS standard, from http://mqtt.org/.

1299. (2020), MQTT, from https://en.wikipedia.org/wiki/MQTT.

1300. (2020), Constrained Application Protocol, from https://en.wikipedia.org/wiki/Constrained_Application_Protocol.

1301. CoAP, from https://coap.technology/.

1302. (2019), 6LoWPAN, from https://en.wikipedia.org/wiki/6LoWPAN.

1303. (2020), Edge computing, from https://en.wikipedia.org/wiki/Edge_computing.

1304. (2020), ANT (network), from https://en.wikipedia.org/wiki/ANT_(network).

1305. (2019), OMA LWM2M, from https://en.wikipedia.org/wiki/OMA_LWM2M.

1306. (2020), IoT Standards and Protocols, from https://www.postscapes.com/internet-of-things-protocols/.

1307. (2020), Google Fuchsia, from https://en.wikipedia.org/wiki/Google_Fuchsia.

1308. Rita Sharma, Top 10 Challenges Enterprises Face in IoT Implementation, from https://www.finoit.com/blog/enterprise-challenges-in-iot/.

1309. OWASP Internet of Things, from https://owasp.org/www-project-internet-of-things/#tab=IoT_Attack_Surface_Areas.

1310. Internet of Things (IoT) Threats, from https://appsec-labs.com/iot_threats/#toggle-id-5.

1311. (2019), IoT Application Security Challenges and Solutions, from https://www.iotforall.com/iot-application-security/.

1312. Ryan Kh, (2018), Assessing the Severity of SQL Injection Threats to IoT Security, from https://www.smartdatacollective.com/assessing-severity-sql-injection-threats-iot-security/.

1313. Anand Srinivasan, (2017), Understanding SDR-Based Attacks on IoT, from https://datafloq.com/read/understanding-sdr-based-attacks-on-iot/3735.

1314. Nitesh Malviya, IoT Radio Communication Attack, from https://resources.infosecinstitute.com/iot-radio-communication-attack/#gref.

1315. Robert Keim, (2017), Introduction to Software-Defined Radio, from https://www.allaboutcircuits.com/technical-articles/introduction-to-software-defined-radio/.

1316. Rene Millman, (2018), Hackers Could Use Web-based Attacks to Take Over IoT Devices, from https://internetofbusiness.com/hackers-could-use-web-based-attacks-to-take-over-iot-devices/.

1317. (2019), Discovering and Hacking IoT Devices Using Web-Based Attacks, from https://www.netsparker.com/blog/web-security/discovering-hacking-iot-devices-using-web-based-attacks/.

1318. Gunes Acar, Danny Y. Huang, Frank Li, Arvind Narayanan, and Nick Feamster, (2018), Fast Web-based Attacks to Discover and Control IoT Devices, from https://freedom-to-tinker.com/2018/06/21/fast-web-based-attacks-to-discover-and-control-iot-devices/.

1319. Gunes Acar, Danny Huang, Frank Li, Arvind Narayanan, and Nick Feamster, Web-based Attacks on Local IoT Devices, from https://conferences.sigcomm.org/sigcomm/2018/files/slides/iot/paper_3.1.pdf.

1320. Margaret Rouse, (2008), DNS Rebinding Attack, from https://searchsecurity.techtarget.com/definition/DNS-rebinding-attack.

1321. Kobus Marneweck, (2019), The Role of Physical Security in IoT, from https://community.arm.com/iot/b/blog/posts/the-role-of-physical-security-in-iot.

1322. Shivam Bhasin and Debdeep Mukhopadhyay, (2016), Fault Injection Attacks, from https://pdfs.semanticscholar.org/0ae1/a6e055383e64011fa639e42f9294d11c3639.pdf.

1323. Hezam Akram Abdul-Ghani, Dimitri Konstantas, and Mohammed Mahyoub, (2018), A Comprehensive IoT Attacks Survey Based on a Building-blocked Reference Model, from https://thesai.org/Downloads/Volume9No3/Paper_49-A_Comprehensive_IoT_Attacks_Survey.pdf.

1324. (2019), 2019 Internet Security Threat Report, from https://www.symantec.com/security-center/threat-report.

1325. Karen Taylor, Mark Steedman, Amen Sanghera, and Matthew Thaxter, (2018), Medtech and the Internet of Medical Things, from https://www2.deloitte.com/content/dam/Deloitte/global/Documents/Life-Sciences-Health-Care/gx-lshc-medtech-iomt-brochure.pdf.

1326. Dr Leonie Maria Tanczer, Dr Ine Steenmans, Dr Irina Brass, and Dr Madeline Carr, (2018), Networked World Risks and Opportunities in the Internet of Things, from https://discovery.ucl.ac.uk/id/eprint/10063068/1/InterconnectedWorld2018.pdf.

1327. Smart Home Statistics, from https://ipropertymanagement.com/research/iot-statistics.

1328. Cyrill Brunschwiler, (2016), Software Defined Radio (SDR) and Decoding On-off Keying (OOK), from https://blog.compass-security.com/2016/09/software-defied-radio-sdr-and-decoding-on-off-keying-ook/.

1329. Aaron Guzman and Aditya Gupta, Defining Firmware Analysis Methodology, from https://play.google.com/books/reader?id=rEFPDwAAQBAJ&hl=en&pg=GBS.PA73.w.2.0.0.

1330. OWASP Internet of Things, from https://owasp.org/www-project-internet-of-things/#Things_to_check_for_once_the_file_system_is_mounted_or_extracted.

1331. Industrial IoT: Threats and Countermeasures, from https://www.rambus.com/iot/industrial-iot/.

1332. Internet of Things (IoT) security: 9 ways you can help protect yourself, from https://us.norton.com/internetsecurity-iot-securing-the-internet-of-things.html.

1333. Cujo AI, (2018), Five Key Security Tips to Avoid an IoT Hack, from https://www.helpnetsecurity.com/2018/08/14/avoid-iot-hack/.

1334. Common Attacks on IoT Devices, from https://elinux.org/images/f/f8/Common-Attacks-on-IoT-Devices-Christina-Quast.pdf.

1335. Lyndon Sutherland, (2017), The Weaponization of IoT Devices, from https://www.ibm.com/downloads/cas/6MLEALKV.

1336. Jeff Day, Roger Shepherd, Paul Kearney and Richard Storer, (2018), Best Practice Guides, from https://www.iotsecurityfoundation.org/wp-content/uploads/2019/03/Best-Practice-Guides-Release-1.2.1.pdf.

1337. OWASP's Top 10 IoT Vulnerabilities, from https://www.deviceauthority.com/blog/owasp-s-top-10-iot-vulnerabilities.

1338. Fredric Paul, (2019), Top 10 IoT vulnerabilities, from https://www.networkworld.com/article/3332032/top-10-iot-vulnerabilities.html.

1339. (2016), IoT Framework Assessment, from https://www.owasp.org/index.php/IoT_Framework_Assessment.

1340. Calum McClelland, (2019), IoT Device Management: What is it and Why Do You Need it?, from https://www.iotforall.com/what-is-iot-device-management/.

1341. (2019), Using Oracle Internet of Things Asset Monitoring Cloud Service, from https://docs.oracle.com/en/cloud/saas/iot-asset-cloud/iotaa/operations-center.html#GUID-25C1A99C-850D-40E6-864D-04CC6CE295C4.

1342. (2020), Operational Technology, from https://en.wikipedia.org/wiki/Operational_Technology.

1343. Lauren Horwitz, OT networks and IT networks are closely intertwined, from https://www.cisco.com/c/en/us/products/security/ot-networks.html.

1344. Operational Technology (OT) – Definitions and Differences with IT, from https://www.i-scoop.eu/industry-4-0/operational-technology-ot/.

1345. Graham Williamson, (2015), OT, ICS, SCADA – What's the difference?, from https://www.kuppingercole.com/blog/williamson/ot-ics-scada-whats-the-difference.

1346. OT Definition - Operational Technology, from http://www.commtunnel.com/ot-definition/.

1347. About Industrial Networks, from https://www.oreilly.com/library/view/industrial-network-security/9780124201149/B9780124201149000022/B9780124201149000022.xhtml#B9780124201149000022.

1348. Mohamed Babikir, (2018), Convergence of IT and OT in Energy and Manufacturing, from https://www.digitalistmag.com/cio-knowledge/2018/11/05/convergence-of-it-ot-in-energy-manufacturing-06192743.

1349. Tim Sowell, (2015), OT/IT Convergence "What does it mean in the Industrial World?", from http://operationalevolution.blogspot.com/2015/02/otit-convergence-what-does-it-mean-in.html.

1350. Bridging the Gap Between Operational Technology and Information Technology, from https://www.avnet.com/wps/wcm/connect/onesite/90fb068d-33a4-4039-970e-91bea619456f/pa-eurotechot-it-whitepaper-inc0364043-0416-en.pdf?MOD=AJPERES&CVID=lFRXUrV&id=1489688438797.

1351. Beginners: What is Industrial IoT (IIoT), from https://www.youtube.com/watch?v=6MN0xRJ3yzE.

1352. The Purdue model for Industrial control systems, from https://subscription.packtpub.com/book/networking_and_servers/9781788395151/1/ch01lvl1sec10/the-purdue-model-for-industrial-control-systems.

1353. (2019), Blueprint for Securing Industrial Control Systems, from https://www.checkpoint.com/downloads/products/cp-industrial-control-ics-security-blueprint.pdf.

1354. Ethernet-to-the-Factory 1.2 Design and Implementation Guide, from https://www.cisco.com/c/en/us/td/docs/solutions/Verticals/EttF/EttFDIG/ch2_EttF.html.

1355. Rick Peters, (2019), Key Findings on the State of Operational Technology and Cybersecurity, from https://www.csoonline.com/article/3392579/key-findings-on-the-state-of-operational-technology-and-cybersecurity.html#:~:targetText=Cybersecurity%20Risks%20for%20Operational%20Technology&targetText=The%20most%20common%20types%20of,spyware%2C%20and%20mobile%20security%20breaches.

1356. (2018), Identifying the Risks to Operational Technology, from https://www.horn-it.com/2018/09/identifying-the-risks-to-operational-technology/.

1357. (2018), Why a Unified Approach to IT and OT Network Security is Critical, from https://www.infosecurityeurope.com/__novadocuments/480018?v=636628570667070000.

1358. (2019), The IIoT Attack Surface: Threats and Security Solutions, from https://www.trendmicro.com/vinfo/us/security/news/internet-of-things/-the-iiot-attack-surface-threats-and-security-solutions.

1359. Keith Stouffer, Victoria Pillitteri, Suzanne Lightman, Marshall Abrams, and Adam Hahn, (2015), Guide to Industrial Control Systems (ICS) Security, from https://nvlpubs.nist.gov/nistpubs/specialpublications/nist.sp.800-82r2.pdf.

1360. Industrial Control System, from https://www.trendmicro.com/vinfo/in/security/definition/industrial-control-system.

1361. Arthur Gervais, (2012), Security Analysis of Industrial Control Systems, from http://nordsecmob.aalto.fi/en/publications/theses_2012/gervais-arthur_thesis.pdf.

1362. Pascal Ackerman, The Industrial control system architecture, from https://www.oreilly.com/library/view/industrial-cybersecurity/9781788395151/4e1de95d-dcb4-44bd-b845-4706bc4c4682.xhtml.

1363. Distributed control system, from https://www.oreilly.com/library/view/industrial-cybersecurity/9781788395151/9116dbe5-4ade-43fc-9ea4-1adcaa6cdac1.xhtml.

1364. Everything You Need to Know About Distributed Control System, from https://www.elprocus.com/distributed-control-system-features-and-elements/.

1365. Pascal Ackerman, Supervisory Control and Data Acquisition, from https://www.oreilly.com/library/view/industrial-cybersecurity/9781788395151/b99736a0-fe9d-4b70-9ea8-b6a76142ffd0.xhtml.

1366. SCADA Overview, from https://research.aurainfosec.io/scada-penetration-testing/#scada-overview.

1367. What is a PLC System – Different Types of PLCs with Applications, from https://www.elprocus.com/programmable-logic-controllers-and-types-of-plcs/.

1368. PLC: Industrial Applications of Programmable Logic Controller, from https://www.mobileautomation.com.au/plc-industrial-application/.

1369. Basic Process Control Systems, from http://stiautomation.net/basic-process-control-systems/.

1370. (2015), Basic Process Control System (BPCS), from http://thamilkash.blogspot.com/2015/06/identify-basic-process-control-system.html.

1371. Process Automation Protocols, from https://www.oreilly.com/library/view/industrial-cybersecurity/9781788395151/f1cac712-ca9d-4361-a4a6-715a4bb1d6b6.xhtml.

1372. Miguel Herrero Collantes and Antonio López Padilla, (2015), Protocols and Network Security in ICS Infrastructures, from https://www.incibe.es/extfrontinteco/img/File/intecocert/ManualesGuias/incibe_protocol_net_security_ics.pdf.

1373. (2016), Communication Network Dependencies for ICS/SCADA Systems, from https://www.enisa.europa.eu/publications/ics-scada-dependencies/at_download/fullReport.

1374. (2018), Industrial Protocols, from https://uploads-ssl.webflow.com/57b990500c7bdb623821b928/5ab757d71f3370e024d6184c_ProtocolUpdateDataSheet.pdf.

1375. Operational Technology and Security, from http://trustcentral.com/use-cases/operational-technology-ot-and-iiot/.

1376. Pascal Ackerman, Communication Protocols in The Enterprise Zone, from https://www.oreilly.com/library/view/industrial-cybersecurity/9781788395151/0af70163-afb9-40d1-8ec2-fdfe65e2d3b1.xhtml.

1377. Glenn Schatz, (2016), Wireless IoT Network Protocols, from https://www.link-labs.com/blog/complete-list-iot-network-protocols.

1378. (2019), List of Automation Protocols, from https://en.wikipedia.org/wiki/List_of_automation_protocols.

1379. LoRa Network Protocol and Long Range Wireless IoT, from https://www.postscapes.com/long-range-wireless-iot-protocol-lora/.

1380. ICS Protocols, from https://resources.infosecinstitute.com/category/certifications-training/ics-scada/ics-protocols/#gref.

1381. Sercos II, from https://www.kunbus.com/sercos-ii.html.

1382. S7 Communication (S7comm), from https://wiki.wireshark.org/S7comm.

1383. Siemens S7 Protocol, from http://settimino.sourceforge.net/s7_protocol.html.

1384. (2015), Cyber Savvy: Securing Operational Technology Assets, from https://www.pwc.com.au/pdf/securing-operational-technology-assets.pdf.

1385. (2016), Critical Infrastructure and SCADA/ICS Cybersecurity Vulnerabilities and Threats, from https://www.checkpoint.com/downloads/products/top-10-cybersecurity-vulnerabilities-threat-for-critical-infrastructure-scada-ics.pdf.

1386. Top 10 Operational Technology Security Threats, from https://www.bayshorenetworks.com/top-10-ot-threats.

1387. Eduard Kovacs, (2018), Vulnerability Exposes Rockwell Controllers to DoS Attacks, from https://www.securityweek.com/vulnerability-exposes-rockwell-controllers-dos-attacks.

1388. Brian Gorenc and Fritz Sands, Hacker Machine Interface, from https://documents.trendmicro.com/assets/wp/wp-hacker-machine-interface.pdf.

1389. (2019), Singapore's Operational Technology Cybersecurity Masterplan 2019, from https://www.csa.gov.sg/~/media/csa/documents/publications/ot_masterplan/csa_ot_masterplan.pdf.

1390. Galina Antova, (2017), Rethinking Vulnerability Disclosures in Industrial Control Systems, from https://www.darkreading.com/threat-intelligence/rethinking-vulnerability-disclosures-in-industrial-control-systems/a/d-id/1327986.

1391. Irfan Shakeel, (2016), Destructive Security Flaws in Industrial Control Systems, from https://resources.infosecinstitute.com/destructive-security-flaws-industrial-control-systems/#gref.

1392. (2018), Side-Channel Attacks Put Critical Infrastructure at Risk, from https://www.icscybersecurityconference.com/side-channel-attacks-put-critical-infrastructure-at-risk/.

1393. Eduard Kovacs, (2018), ICS Devices Vulnerable to Side-Channel Attacks: Researcher, from https://www.securityweek.com/ics-devices-vulnerable-side-channel-attacks-researcher.

1394. Dr. Siv Hilde Houmb, (2018), How to Hack Programmable Logic Controllers, from https://www.controldesign.com/articles/2018/how-to-hack-programmable-logic-controllers/.

1395. Ali Abbasi and Majid Hashemi, (2016), Ghost in the PLC: Designing an Undetectable Programmable Logic Controller Rootkit via Pin Control Attack, from https://research.utwente.nl/en/publications/ghost-in-the-plc-designing-an-undetectable-programmable-logic-con.

1396. (2019), Attacks Against Industrial Machines via Vulnerable Radio Remote Controllers: Security Analysis and Recommendations, from https://www.trendmicro.com/vinfo/us/security/news/vulnerabilities-and-exploits/attacks-against-industrial-machines-via-vulnerable-radio-remote-controllers-security-analysis-and-recommendations.

1397. Bruce Sussman, (2019), Industrial Cybersecurity: RF Vulnerability, from https://www.secureworldexpo.com/industry-news/industrial-cybersecurity-risk-study.

1398. (2018), An Introduction to Operational Technology and it's Security: 5 Key Facts, from https://www.vsec.infinigate.co.uk/blog/operational-technology-security-ransomware-threats.

1399. Marcel Kisch, (2017), What Do Recent Attacks Mean for OT Network Security?, from https://securityintelligence.com/what-do-recent-attacks-mean-for-ot-network-security/.

1400. (2019), MegaCortex Ransomware Spotted Attacking Enterprise Networks, from https://www.trendmicro.com/vinfo/pl/security/news/cybercrime-and-digital-threats/megacortex-ransomware-spotted-attacking-enterprise-networks.

1401. Lawrence Abrams, (2019), Elusive MegaCortex Ransomware Found - Here is What We Know, from https://www.bleepingcomputer.com/news/security/elusive-megacortex-ransomware-found-here-is-what-we-know/.

1402. Pieter Arntz, (2019), MegaCortex Continues Trend of Targeted Ransomware Attacks, from https://blog.malwarebytes.com/threat-spotlight/2019/06/megacortex-continues-trend-of-targeted-ransomware-attacks/.

1403. Michelle Netten, (2019), MegaCortex Ransomware Worsens — Hackers Change Users' Passwords & Make Blackmail Demands, from https://virsec.com/megacortex-ransomware-worsens-hackers-change-users-passwords-and-make-blackmail-demands/.

1404. Jasper Manuel and Joie Salvio, (2019), LockerGoga: Ransomware Targeting Critical Infrastructure, from https://www.fortinet.com/blog/threat-research/lockergoga-ransomeware-targeting-critical-infrastructure.html.

1405. (2019), What You Need to Know About the LockerGoga Ransomware, from https://www.trendmicro.com/vinfo/us/security/news/cyber-attacks/what-you-need-to-know-about-the-lockergoga-ransomware/.

1406. (2019), Targeted Ransomware: Proliferating Menace Threatens Organizations, from https://www.symantec.com/blogs/threat-intelligence/targeted-ransomware-threat.

1407. (2017), Hardware Hacking in Industrial Control Systems,from https://www.incibe-cert.es/en/blog/hardware-hacking-industrial-control-system.

1408. (2017), SCADA Hacking: Hacking the Schneider Electric TM221 Modicon PLC using modbus-cli, from https://www.hackers-arise.com/post/2017/03/28/scada-hacking-hacking-the-schneider-electric-tm221-modicon-plc-using-modbus-cli.

1409. (2018), SCADA Hacking: Exploiting SCADA/ICS Systems with the Command Line Tool, modbus-cli, from https://www.hackers-arise.com/post/2018/03/22/scada-hacking-exploiting-scadaics-systems-with-the-command-line-tool-modbus-cli.

1410. William Grove, (2019), OT Networks Saw Attacks Continue to Rise in 2018, from https://blog.skyboxsecurity.com/ot_networks_2018_threats/.

1411. An Executive Guide to Cyber Security for Operational Technology, from https://www.ge.com/fr/sites/www.ge.com.fr/files/an-executive-guide-to-cyber-security-for-operational-technology-whitepaper.pdf.

1412. Dr. J.M. Ceron, Dr. J.J.Chromik, Dr. J.J.C. Santanna and Prof. dr. ir. A. Pras, (2019), Online Discoverability and Vulnerabilities of ICS/SCADA Devices in the Netherlands, from https://www.thehaguesecuritydelta.com/media/com_hsd/report/250/document/wodc-report-scada-final.pdf.

1413. (2011), Common Cybersecurity Vulnerabilities in Industrial Control Systems, from https://www.us-cert.gov/sites/default/files/recommended_practices/DHS_Common_Cybersecurity_Vulnerabilities_ICS_2010.pdf.

1414. Adrian Booth, Aman Dhingra, Sven Heiligtag, Mahir Nayfeh, and Daniel Wallance, (2019), Critical Infrastructure Companies and the Global Cybersecurity Threat, from https://www.mckinsey.com/business-functions/risk/our-insights/critical-infrastructure-companies-and-the-global-cybersecurity-threat.

1415. Lauren Gibbons Paul, (2018), Making Sense of the ICS Cybersecurity Market, from https://www.automationworld.com/home/article/13318353/making-sense-of-the-ics-cybersecurity-market.

1416. Open Source Honeypots That Detect Threats for Free, from https://www.smokescreen.io/practical-honeypots-a-list-of-open-source-deception-tools-that-detect-threats-for-free/.

Module 19: Cloud Computing

1417. (2013), Cloud Computing Vulnerability Incidents: A Statistical Overview, from https://cloudsecurityalliance.org/download/cloud-computing-vulnerability-incidents-a-statistical-overview/.

1418. (2013), Introduction to Cloud Computing, from https://www.slideshare.net/ProfEdge/introduction-to-cloud-computing-23970527.

1419. Alok Tripathi and Abhinav Mishra, (2011), Cloud Computing Security Considerations, from https://www.semanticscholar.org/paper/Cloud-computing-security-considerations-Tripathi-Mishra/fd710d62f8db9621d97ab00acf1bb8e8d28e06b2.

1420. Kazi Zunnurhain and Susan V. Vrbsky, Security Attacks and Solutions in Clouds, from http://salsahpc.indiana.edu/CloudCom2010/Poster/cloudcom2010_submission_98.pdf.

1421. Chimere Barron, Huiming Yu and Justin Zhan (2013), Cloud Computing Security Case Studies and Research, from http://www.iaeng.org/publication/WCE2013/WCE2013_pp1287-1291.pdf.

1422. Keiko Hashizume, David G Rosado, Eduardo Fernández-Medina and Eduardo B Fernandez (2013), An analysis of security issues for cloud computing, from https://jisajournal.springeropen.com/articles/10.1186/1869-0238-4-5.

1423. Ian Mitchell and John Alcock, Cloud Security The definitive guide to managing risk in the new ICT landscape, from https://www.fujitsu.com/global/Images/WBOC-2-Security.pdf.

1424. Michael Cobb, (2007), What security issues can arise from unsynchronized system clocks?, from https://searchsecurity.techtarget.com/answer/What-security-issues-can-arise-from-unsynchronized-system-clocks.

1425. Network Time Synchronization, from https://www.endruntechnologies.com/network-time-synchronization.htm.

1426. Man in the Cloud (MITC) Attacks, from https://www.imperva.com/docs/HII_Man_In_The_Cloud_Attacks.pdf.

1427. Martin Gontovnikas, (2018), What Is Identity as a Service (IDaaS)?, from https://auth0.com/blog/identity-as-a-service-in-2018/.

1428. Multi-Cloud, from https://avinetworks.com/glossary/multi-cloud/.

1429. (2019), Multicloud, from https://en.wikipedia.org/wiki/Multicloud.

1430. Rich Caldwell, (2019), Pros and Cons of a Multi-Cloud Strategy, from https://centricconsulting.com/blog/pros-and-cons-of-a-multi-cloud-strategy/.

1431. Jignesh Solanki, 6 Multi-Cloud Architecture Designs for an Effective Cloud Strategy, from https://www.simform.com/multi-cloud-architecture/.

1432. (2020), Cloud storage, from https://en.wikipedia.org/wiki/Cloud_storage.

1433. Laxmi Ashrit, What is Cloud Storage – Architecture, Types, Advantages & Disadvantages, from https://electricalfundablog.com/cloud-storage-architecture-types/.

1434. Basic Cloud Storage Architecture Information Technology Essay, from https://www.uniassignment.com/essay-samples/information-technology/basic-cloud-storage-architecture-information-technology-essay.php.

1435. Liz Alton , (2019), 4 Ways AI Is Improving Cloud Computing, from https://community.connection.com/4-ways-ai-is-improving-cloud-computing/.

1436. Vishal Bhatia, The Role of Artificial Intelligence in Cloud Computing, from https://www.goodfirms.co/blog/role-of-ai-in-cloud-computing.

1437. Neeru Jain, (2019), Top 7 Benefits of Using AI in Cloud Computing, from https://www.whizlabs.com/blog/benefits-of-ai-in-cloud-computing/.

1438. Bob O'Donnell, (2016), Virtual Reality and the Cloud Belong Together, from https://www.vox.com/2016/4/6/11585914/virtual-reality-in-the-cloud.

1439. (2019), What is Containers as a service (CaaS)?, from https://www.ibm.com/services/cloud/containers-as-a-service.

1440. Murugiah Souppaya, John Morello, and Karen Scarfone, (2017), Application Container Security Guide, from https://nvlpubs.nist.gov/nistpubs/specialpublications/nist.sp.800-190.pdf.

1441. Sten Pittet, What is a Container?, from https://www.atlassian.com/continuous-delivery/microservices/containers.

1442. Pethuru Raj, Jeeva S. Chelladhurai, and Vinod Singh, (2015), Containerization vs Virtualization – An introduction to Docker, from https://jaxenter.com/containerization-vs-virtualization-docker-introduction-120562.html.

1443. How is containerization different from virtualization?, from https://www.techopedia.com/7/31288/technology-trends/how-is-containerization-different-from-virtualization.

1444. Roderick Bauer, (2018), What's the Diff: VMs vs Containers, from https://www.backblaze.com/blog/vm-vs-containers/.

1445. (2020), Docker (software), from https://en.wikipedia.org/wiki/Docker_(software).

1446. Docker overview, from https://docs.docker.com/engine/docker-overview/.

1447. Docker Containers, from https://www.aquasec.com/wiki/display/containers/Docker+Containers.

1448. Avi, (2019), Docker Architecture and its Components for Beginner, from https://geekflare.com/docker-architecture/.

1449. Docker Architecture, from https://www.aquasec.com/wiki/display/containers/Docker+Architecture.

1450. Swarm mode overview, from https://docs.docker.com/engine/swarm/.

1451. What is Docker Swarm, from https://www.aquasec.com/wiki/display/containers/Docker+Containers#DockerContainers-DOCKERSWARM….

1452. (2019), Designing a Microservices Architecture with Docker Containers, from https://www.sumologic.com/insight/microservices-architecture-docker-containers/.

1453. Asad Faizi, (2019), Microservices Orchestration with Kubernetes, from https://medium.com/faun/microservices-orchestration-with-kubernetes-1cbb737cfa46.

1454. Mark Church, Marlon Ruiz, Andrew Seifert, and Trapier Marshall, Docker Swarm Reference Architecture: Exploring Scalable, Portable Docker Container Networks, from https://success.docker.com/article/networking.

1455. (2018), Docker Networking, from https://github.com/kyhau/docker-notebook/blob/master/docker-networking.md.

1456. Saurabh Kulshrestha, (2018), Docker Networking - Explore How Containers Communicate With Each Other, from https://medium.com/edureka/docker-networking-1a7d65e89013.

1457. Isaac Eldridge, (2018), What Is Container Orchestration?, from https://blog.newrelic.com/engineering/container-orchestration-explained/.

1458. Container Orchestration, from https://avinetworks.com/glossary/container-orchestration/.

1459. (2019), What is Kubernetes, from https://kubernetes.io/docs/concepts/overview/what-is-kubernetes/.

1460. Kubernetes Architecture 101, from https://www.aquasec.com/wiki/display/containers/Kubernetes+Architecture+101.

1461. (2020), Kubernetes Components, from https://kubernetes.io/docs/concepts/overview/components/.

1462. Guillermo Velez, (2019), Kubernetes vs. Docker: A Primer, from https://containerjournal.com/topics/container-ecosystems/kubernetes-vs-docker-a-primer/.

1463. Jim Armstrong, Docker And Kubernetes? I thought you were competitors?, from https://www.docker.com/blog/top-questions-docker-kubernetes-competitors-or-together/.

1464. Amir Jerbi, (2017), 8 Docker security rules to live by, from https://www.infoworld.com/article/3154711/8-docker-security-rules-to-live-by.html.

1465. (2018), Security Challenges Related to Containers, from https://www.ariacybersecurity.com/container-security-challenges-blog/.

1466. Christopher Tozzi,(2018) , 3 Container Security Advantages and 3 Security Challenges, from https://containerjournal.com/topics/container-security/3-container-security-advantages-and-3-security-challenges/.

1467. Paul Castro, Vatche Ishakian, Vinod Muthusamy, and Aleksander Slominski, (2019), The Rise of Serverless Computing, from https://cacm.acm.org/magazines/2019/12/241054-the-rise-of-serverless-computing/fulltext.

1468. (2020), Serverless Computing, from https://en.wikipedia.org/wiki/Serverless_computing.

1469. David Prothero, Serverless Architecture, from https://www.twilio.com/docs/glossary/what-is-serverless-architecture.

1470. Why use Serverless Computing? | Pros and Cons of Serverless, from https://www.cloudflare.com/learning/serverless/why-use-serverless/.

1471. Jignesh Solanki, Serverless Architecture: A Comprehensive Guide, from https://www.simform.com/serverless-architecture-guide/.

1472. (2014), Cloud Top 10 Security Risks, from https://www.owasp.org/index.php/Category:OWASP_Cloud_%E2%80%90_10_Project.

1473. Shankar Babu, Chebrolu, Vinay Bansal, and Pankaj Telang, Top 10 Cloud Risks That Will Keep You Awake at Night, from https://www.owasp.org/images/4/47/Cloud-Top10-Security-Risks.pdf.

1474. OWASP Top 10 (2017) Interpretation for Serverless, from https://www.owasp.org/images/5/5c/OWASP-Top-10-Serverless-Interpretation-en.pdf.

1475. Lance Whitney, (2019), How to Prevent the Top 11 Threats in Cloud Computing, from https://www.techrepublic.com/article/how-to-prevent-the-top-11-threats-in-cloud-computing/.

1476. Chester Avey, (2019), 7 Key Cybersecurity Threats to Cloud Computing, from https://cloudacademy.com/blog/key-cybersecurity-threats-to-cloud-computing/.

1477. Rakesh Soni, (2019), The Rise of Cloud Computing Threats: How to protect your cloud customers from security risks, from https://customerthink.com/the-rise-of-cloud-computing-threats-how-to-protect-your-cloud-customers-from-security-risks/.

1478. (2019), Container Security: Examining Potential Threats to the Container Environment, from https://www.trendmicro.com/vinfo/us/security/news/security-technology/container-security-examining-potential-threats-to-the-container-environment.

1479. Container Vulnerabilities and Threats, from https://www.aquasec.com/wiki/display/containers/Container+Vulnerabilities+and+Threats.

1480. (2019), Cloud Container Vulnerabilities Soar, According to Report, from https://www.globenewswire.com/news-release/2019/07/24/1886967/0/en/Cloud-Container-Vulnerabilities-Soar-According-to-Report.html.

1481. Anurag Kahol, (2019), Beware the man in the cloud: How to protect against a new breed of cyberattack, from https://www.helpnetsecurity.com/2019/01/21/mitc-attack/.

1482. Adrian Nish and Tom Rowles, (2017), APT10 - OPERATION CLOUD HOPPER, from https://baesystemsai.blogspot.com/2017/04/apt10-operation-cloud-hopper_3.html.

1483. Nathaniel Richmond, (2019), Operation Cloud Hopper Case Study, from https://insights.sei.cmu.edu/sei_blog/2019/03/operation-cloud-hopper-case-study.html.

1484. Jeremy Kirk, (2019), Cloud Hopper: Major Cloud Services Victims Named, from https://www.bankinfosecurity.com/cloud-hopper-major-cloud-services-victims-named-a-12695.

1485. (2018), Cryptojacking Attacks - Securonix Security Advisory (SSA), from https://www.securonix.com/web/wp-content/uploads/2018/06/cryptojacking_security_advisory.pdf.

1486. Charlie Osborne, (2018), Cryptojacking Attacks Surge Against Enterprise Cloud Environments, from https://www.zdnet.com/article/cryptojacking-attacks-surge-against-enterprise-cloud-environments/.

1487. Trenton Baker, (2018), Mobile and Cloud Cryptojacking Skyrockets, from https://www.keepitsafe.com/blog/post/mobile-and-cloud-cryptojacking-skyrockets/.

1488. Tara Seals, (2019), 'Cloudborne' IaaS Attack Allows Persistent Backdoors in the Cloud, from https://threatpost.com/cloudborne-iaas-attack-cloud/142223/.

1489. Rene Millman, (2019), Bare metal flaw allows hackers to put backdoors into cloud servers, from https://www.cloudpro.co.uk/it-infrastructure/security/7961/bare-metal-flaw-allows-hackers-to-put-backdoors-into-cloud-servers.

1490. Maria Deutscher, New Cloudborne vulnerability exposes cloud servers to potential hacking, from https://siliconangle.com/2019/02/26/new-cloudborne-vulnerability-potentially-exposes-cloud-servers-hacking/.

1491. Kelly Sheridan, (2019), Cloudborne: Bare-Metal Cloud Servers Vulnerable to Attack, from https://www.darkreading.com/cloud/cloudborne-bare-metal-cloud-servers-vulnerable-to-attack/d/d-id/1333969.

1492. Aditya K Sood and Rehan Jalil, (2018), Cloudifying Threats—Understanding Cloud App Attacks and Defenses, from https://www.isaca.org/Journal/archives/2018/Volume-1/Pages/cloudifying-threats-understanding-cloud-app-attacks-and-defenses.aspx?utm_referrer=.

1493. Metadata Spoofing Attack, from https://books.google.as/books?id=x4e6BQAAQBAJ&pg=PA182&lpg=PA182&dq=Cloud+Malware+Injection+Attack&source=bl&ots=xv-egwmdU9&sig=ACfU3U1HzG8MIciUu-SwIQzFsLLjGUlvBw&hl=en&sa=X&ved=2ahUKEwjSwsmigoXmAhWY63MBHe64A944KBDoATAFegQICBAB#v=onepage&q=metadata%20spoofing&f=false.

1494. What is an HTTP flood DDoS attack?, from https://www.cloudflare.com/learning/ddos/http-flood-ddos-attack/.

1495. (2017), Detection of HTTP Flooding Attacks in Cloud Using Dynamic Entropy Method, from https://www.researchgate.net/publication/321185428_Detection_of_HTTP_Flooding_Attacks_in_Cloud_Using_Dynamic_Entropy_Method.

1496. Anna Bryk, (2018), Cloud Computing: A New Vector for Cyber Attacks, from https://www.apriorit.com/dev-blog/523-cloud-computing-cyber-attacks.

1497. (2019), Top 5 Cloud Computing Security Issues; and How they are used by Hackers, from https://www.cloudmanagementinsider.com/top-5-cloud-computing-security-issues-and-strategies-used-by-hackers/.

1498. Warwick Ashford, (2018), Hackers Increasingly Targeting Cloud Infrastructure, from https://www.computerweekly.com/news/252444716/Hackers-increasingly-targeting-cloud-infrastructure.

1499. (2018), A Practical Guide to Testing the Security of Amazon Web Services (Part 1: AWS S3), from https://blog.mindedsecurity.com/2018/09/a-practical-guide-to-testing-security.html.

1500. Working with Amazon S3 Buckets, from https://docs.aws.amazon.com/AmazonS3/latest/dev/UsingBucket.html.

1501. Rohan Chavan, (2019), Finding and Testing MisConfigured S3 Buckets, from https://medium.com/@rohanchavan/finding-and-testing-misconfigured-s3-buckets-d77992c4b5cd.

1502. Rmorril, (2012), Google hacking Amazon Web Services Cloud front and S3, from https://it.toolbox.com/blogs/rmorril/google-hacking-amazon-web-services-cloud-front-and-s3-011613.

1503. Matthew Palmer, How Does Kubernetes Use etcd?, from https://matthewpalmer.net/kubernetes-app-developer/articles/how-does-kubernetes-use-etcd.html.

1504. Spencer Gietzen, Enumerating AWS Roles through 'AssumeRole', from https://rhinosecuritylabs.com/aws/assume-worst-aws-assume-role-enumeration/.

1505. Jay Jani, (2019), S3 Bucket Misconfiguration: From Basics to Pawn, from https://bugbountypoc.com/s3-bucket-misconfiguration-from-basics-to-pawn/.

1506. Spencer Gietzen, Cloud Breach: Compromising AWS IAM Credentials, from https://rhinosecuritylabs.com/aws/aws-iam-credentials-get-compromised/.

1507. Vince Lujan, (2017), What is AWS IAM?, from https://jumpcloud.com/blog/what-is-aws-iam/.

1508. Vitaly Simonovich and Ori Nakar, (2019), Hundreds of Vulnerable Docker Hosts Exploited by Cryptocurrency Miners, from https://www.imperva.com/blog/hundreds-of-vulnerable-docker-hosts-exploited-by-cryptocurrency-miners/.

1509. Riyaz Walikar, (2019), Getting Shell and Data Access in AWS by Chaining Vulnerabilities, from https://blog.appsecco.com/getting-shell-and-data-access-in-aws-by-chaining-vulnerabilities-7630fa57c7ed.

1510. Riyaz Walikar, (2019), An SSRF, Privileged AWS Keys and the Capital One Breach , from https://blog.appsecco.com/an-ssrf-privileged-aws-keys-and-the-capital-one-breach-4c3c2cded3af.

1511. Spencer Gietzen, AWS IAM Privilege Escalation – Methods and Mitigation, from https://rhinosecuritylabs.com/aws/aws-privilege-escalation-methods-mitigation/.

1512. Setu, (2018), What is AWS Post exploitation?, from https://cloudsecops.com/aws-post-exploitation-part-1/.

1513. Spencer Gietzen, Google Cloud Platform (GCP) Bucket Enumeration and Privilege Escalation, from https://rhinosecuritylabs.com/gcp/google-cloud-platform-gcp-bucket-enumeration/.

1514. Pawel Rzepa, (2018), Playing with CloudGoat part 2: Fooling AWS CloudTrail and Getting Persistent Access, from https://medium.com/@rzepsky/playing-with-cloudgoat-part-2-fooling-cloudtrail-and-getting-persistence-access-6a1257bb3f7c.

1515. (2019), Cloud Computing Security Considerations, from https://www.cyber.gov.au/publications/cloud-computing-security-considerations.

1516. Top 6 Considerations for Cloud Security and Data Protection, from https://searchstorage.techtarget.com/IronMountainCloud/Top-6-Considerations-For-Cloud-Security-and-Data-Protection.

1517. (2018), Moving to the Cloud – Cloud Security Considerations, from https://cloudcheckr.com/cloud-security/moving-cloud-security/.

1518. Gerry Grealish, Six Key Security Considerations for Responsible Cloud Migration, from https://www.symantec.com/content/dam/symantec/docs/data-sheets/six-key-considerations-for-responsible-cloud-migration-en.pdf.

1519. Cynthia Harvey, (2017), Cloud Security: 11 Best Practices, from https://www.esecurityplanet.com/cloud/cloud-security-best-practices.html.

1520. Jason Meilleur, (2019), The Growing Dangers of Cyber Attacks and the Need for Cloud Security, from https://www.360visibility.com/the-growing-dangers-of-cyber-attacks-and-the-need-for-cloud-security/.

1521. (2019), 19 Cloud Security Best Practices for 2019, from https://securingtomorrow.mcafee.com/blogs/enterprise/cloud-security/top-19-cloud-security-best-practices/.

1522. Matt Miller, (2018), Cloud Security Best Practices, from https://www.beyondtrust.com/blog/entry/cloud-security-best-practices.

1523. Aaron Small, (2019), Kubernetes Security Assessment, from https://github.com/kubernetes/community/blob/master/wg-security-audit/findings/Kubernetes%20Final%20Report.pdf.

1524. Aaron Small, (2019), Kubernetes Threat Model, from https://github.com/kubernetes/community/blob/master/wg-security-audit/findings/Kubernetes%20Threat%20Model.pdf.

1525. Rani Osnat, (2018), Top Docker Security Best Practices, from https://blog.aquasec.com/docker-security-best-practices.

1526. (2017), Major Risks for Core Components of Container Technologies and their Countermeasures NIST SP 800 190, from https://www.hack2secure.com/blogs/major-risks-for-core-components-of-container-technologies-and-their-countermeasures-nist-sp-800-190.

1527. Devdatta Mulgund, (2019), 8 Best Practices for Application Container Security, from https://securityintelligence.com/posts/8-best-practices-for-application-container-security/.

1528. Docker_Security_Cheat_Sheet.md, from https://github.com/OWASP/CheatSheetSeries/blob/master/cheatsheets/Docker_Security_Cheat_Sheet.md.

1529. Jeff Hale, (2019), Top 20 Docker Security Tips, from https://towardsdatascience.com/top-20-docker-security-tips-81c41dd06f57.

1530. (2019), Understanding Docker Container Escapes, from https://blog.trailofbits.com/2019/07/19/understanding-docker-container-escapes/.

1531. (2019), Understanding and Preparing for Container security threats, from https://cyware.com/news/understanding-and-preparing-for-container-security-threats-26031da9.

1532. Docker Security Resources, from https://www.aquasec.com/wiki/display/containers/Docker+Security+Resources.

1533. Sean Michael Kerner, (2019), Serverless Cloud Security: How to Secure Serverless Computing, from https://www.esecurityplanet.com/cloud/serverless-cloud-security.html.

1534. Liran Tal and Guy Podjarny, (2019), 10 Serverless Security Best Practices, from https://snyk.io/blog/10-serverless-security-best-practices/.

1535. Michelle Lange, (2019), What Is Zero Trust, and Why Does It Work?, from https://www.comptia.org/blog/what-is-zero-trust-and-why-does-it-work.

1536. Bianca Soare, (2019), What is the Zero Trust Model?, from https://heimdalsecurity.com/blog/what-is-the-zero-trust-model/.

Module 20: Cryptography

1537. (1999), Cracking S/MIME encryption using idle CPU time, from https://securiteam.com/tools/3J5PRQ0PPQ/.

1538. (2001), Announcing the ADVANCED ENCRYPTION STANDARD (AES), from https://csrc.nist.gov/csrc/media/publications/fips/197/final/documents/fips-197.pdf.

1539. John Talbot and Dominic Welsh, (2006), Complexity and Cryptography an introduction, from https://www.cambridge.org/gb/academic/subjects/mathematics/discrete-mathematics-information-theory-and-coding/complexity-and-cryptography-introduction?format=PB&isbn=9780521617710.

1540. Bruce Schneier, Applied Cryptography, Second Edition, from https://www.schneier.com/books/applied_cryptography/.

1541. Josh Ryder, Introduction to Encryption, from https://www.developer.com/tech/article.php/630681.

1542. Digital Certificates, from https://www.bitpipe.com/tlist/Digital-Certificates.html.

1543. Vijay Bollapragada, Mohamed Khalid, and Scott Wainner, IPSec Authentication and Authorization Models, from http://www.ciscopress.com/articles/article.asp?p=421514&seqNum=4%20-%2031k%20-.

1544. SHA (Secure Hash Algorithm), from http://safeexim.safescrypt.com/SafeDoXX_User_Manual.pdf.

1545. PGP Attack FAQ: The asymmetric cipher, from https://www.iusmentis.com/technology/encryption/pgp/pgpattackfaq/asymmetric/.

1546. What is Public-Key Cryptography?, from http://www.x5.net/faqs/crypto/q3.html.

1547. The Heartbleed Bug, from http://heartbleed.com/.

1548. Dr. B. R Gladman, (2000), The Regulation of Investigatory Powers Bill - The Provisions for Government Access to Keys, from https://www.fipr.org/rip/RIPGAKBG.pdf.

1549. YongBin Zhou and DengGuo Feng, Side-Channel Attacks: Ten Years after Its Publication and the Impacts on Cryptographic Module Security Testing, from https://csrc.nist.gov/csrc/media/events/physical-security-testing-workshop/documents/papers/physecpaper19.pdf.

1550. Van Geelkerken F.W.J, (2006), Digital Mixing (MIX nets), from https://www.iusmentis.com/society/privacy/remailers/onionrouting/.

1551. Josh Ryder, (2000), Introduction to Encryption, from https://www.developer.com/tech/article.php/630681/Introduction-to-Encryption.htm.

1552. Yuan Xue, (2009), Digital Signature, from https://tao.truststc.org/Members/yuanxue/network_security/Public resources/lecture12.

1553. Email Security, from https://www.livinginternet.com/e/es.htm.

1554. Hash-based message authentication code, from https://en.wikipedia.org/wiki/Hash-based_message_authentication_code.

1555. Margaret Rouse, (2010), Hash-based Message Authentication Code (HMAC), from http://searchsecurity.techtarget.com/definition/Hash-based-Message-Authentication-Code-HMAC.

1556. How and when do I use HMAC?, from https://security.stackexchange.com/questions/20129/how-and-when-do-i-use-hmac/20301.

1557. Krishna Gehlot, (2015), Message Authentication Code & HMAC, from, https://www.slideshare.net/PRINCEOFSUNCITY/message-authentication-code-hmac.

1558. Shaanan Cohney, Nadia Heninger, Matthew D. Green, (2017), The DUHK Attack, from https://duhkattack.com/.

1559. Gurubaran S, (2017), DUHK Attack allows Hackers to Recover Encryption Keys and Decrypt Communications Passing Over VPN, from https://gbhackers.com/duhk-attack-decrypt-communications-vpn/.

1560. Mohit Kumar, (2017), DUHK Attack Lets Hackers Recover Encryption Key Used in VPNs & Web Sessions, from https://thehackernews.com/2017/10/crack-prng-encryption-keys.html.

1561. Catalin Cimpanu, (2017), DUHK Crypto Attack Recovers Encryption Keys, Exposes VPN Connections, More, from https://www.bleepingcomputer.com/news/security/duhk-crypto-attack-recovers-encryption-keys-exposes-vpn-connections-more/.

1562. John Leyden, Thomas Claburn and Chris Williams, (2017), 'First ever' SHA-1 hash collision calculated. All it took were five clever brains... and 6,610 years of processor time, from https://www.theregister.co.uk/2017/02/23/google_first_sha1_collision/.

1563. Collision attack, https://en.wikipedia.org/wiki/Collision_attack.

1564. Hash Collision Attack, https://privacycanada.net/hash-functions/hash-collision-attack/.

1565. Threefish from https://www.schneier.com/academic/skein/threefish.html.

1566. (2018), Threefish, from https://en.wikipedia.org/wiki/Threefish.

1567. (2017), Threefish Block Cipher, from https://tinycrypt.wordpress.com/2017/01/07/asmcodes-threefish/.

1568. (2019), Serpent (cipher), from https://en.wikipedia.org/wiki/Serpent_(cipher).

1569. Serpent Algorithm, from http://en.kryptotel.net/serpent.html.

1570. (2019), CAST-128, from https://en.wikipedia.org/wiki/CAST-128.

1571. (2019), Camellia (cipher), from https://en.wikipedia.org/wiki/Camellia_(cipher).

1572. (2020), GOST (block cipher), from https://en.wikipedia.org/wiki/GOST_(block_cipher).

1573. Muhammad Iqbal, Yudi Sahputra, and Andysah Putera Utama Siahaan, (2016), The Understanding of GOST Cryptography Technique, from https://www.researchgate.net/publication/308661220_The_Understanding_of_GOST_Cryptography_Technique.

1574. (2019), YAK(cryptography), from https://en.wikipedia.org/wiki/YAK_(cryptography).

1575. Mohsen Toorani, (2015), Cryptanalysis of a robust key agreement based on public key authentication, from https://onlinelibrary.wiley.com/doi/full/10.1002/sec.1373.

1576. Margaret Rouse, Elliptical Curve Cryptography (ECC), from https://searchsecurity.techtarget.com/definition/elliptical-curve-cryptography.

1577. (2018), Elliptic Curve Cryptography, from https://www.keycdn.com/support/elliptic-curve-cryptography.

1578. Maria Korolov and Doug Drinkwater, (2019), What is quantum cryptography? It's no silver bullet, but could improve security, from https://www.csoonline.com/article/3235970/what-is-quantum-cryptography-it-s-no-silver-bullet-but-could-improve-security.html.

1579. (2020), Quantum cryptography, from https://en.wikipedia.org/wiki/Quantum_cryptography.

1580. Shai Halevi, (2011), Homomorphic Encryption, from https://www.iacr.org/conferences/crypto2011/slides/Halevi.pdf.

1581. (2019), What is the purpose of Homomorphic encryption?, from https://crypto.stackexchange.com/questions/62386/what-is-the-purpose-of-homomorphic-encryption.

1582. Casey Crane, (2019), What is Homomorphic Encryption?, from https://www.thesslstore.com/blog/what-is-homomorphic-encryption/.

1583. Hardware-based Encryption Devices, from https://sourcedaddy.com/aplus/hardware-based-encryption-devices.html.

1584. Darril, TPM and HSM Hardware Encryption Devices, from https://blogs.getcertifiedgetahead.com/tpm-hsm-hardware-encryption-devices/.

1585. Hardware Security Modules (HSMs), from https://safenet.gemalto.com/data-encryption/hardware-security-modules-hsms/.

1586. Crypto USB - What is AES 256-Bit Hardware-Based Encryption?, from https://integralmemory.com/faq/what-aes-256-bit-hardware-based-encryption.

1587. (2020), GNU Privacy Guard, from https://en.wikipedia.org/wiki/GNU_Privacy_Guard.

1588. GNUPG (GPG), from http://www.att.es/producto/goanywhere/products/director/encryption/gpg.html.

1589. NanoDano, (2018), GPG Tutorial, from https://www.devdungeon.com/content/gpg-tutorial.

1590. (2019), Web of trust, from https://en.wikipedia.org/wiki/Web_of_trust.

1591. (2019), Related-key attack, from https://en.wikipedia.org/wiki/Related-key_attack.

1592. Alex Biryukov and Dmitry Khovratovich, Related-key Cryptanalysis of the Full AES-192 and AES-256, from https://eprint.iacr.org/2009/317.pdf.

1593. (2019), Padding oracle attack, from https://en.wikipedia.org/wiki/Padding_oracle_attack.

1594. Nimrod Aviram, Sebastian Schinzel, Juraj Somorovsky, Nadia Heninger, Maik Dankel, Jens Steube, Luke Valenta, David Adrian, J. Alex Halderman, Viktor Dukhovni, Emilia Käsper, Shaanan Cohney, Susanne Engels, Christof Paar, and Yuval Shavitt, (2016), The DROWN Attack, from https://drownattack.com/.

1595. (2019), DROWN attack, from https://en.wikipedia.org/wiki/DROWN_attack.